Wideband OFDM System for Indoor Communication at 60 GHz

Von der Fakultät für Mathematik, Naturwissenschaften und Informatik
der Brandenburgischen Technischen Universität Cottbus

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

genehmigte Dissertation

vorgelegt von

Diplom-Ingenieur

Maxim Piz

geboren am 5. 1. 1975 in Czernowitz / Ukraine

Gutachter: Prof. Dr.-Ing. Rolf Kraemer

Gutachter: Prof. Dr.-Ing. Hermann Rohling

Gutachter: Prof. Dr.-Ing. Heinrich Theodor Vierhaus

Tag der mündlichen Prüfung: 16. 12. 2010

This work has been done at the Leibniz Institute IHP Microelectronics in Frankfurt (Oder) and is based on contributions to the German WIGWAM and EASY-A project. These projects have been funded by the German Federal Ministry of Education and Research (BMBF). The thesis has been submitted at the Brandenburgische Technische Universität Cottbus.

Acknowledgements

I wish to express my utmost gratitude to my supervisors, Prof. Dr. Rolf Kraemer and Dr. Eckhard Grass, who helped me with their invaluable assistance, support and guidance of the work and their careful review of the manuscript. Without my supervisors, this work would not have been possible.

Furthermore, I very thankfully acknowledge Prof. Dr. Hermann Rohling and Prof. Dr. Heinrich Theodor Vierhaus for their thorough reviews and profound suggestions for an improvement of the manuscript.

In addition, my special thanks go to Prof. Dr. Jörg Nolte, Dean of the Faculty of Mathematics, Sciences and Computer Sciences at the University of Cottbus, who directed the defence of my work in a very friendly and pleasant way.

Further acknowledgements go to Dr. Frank Herzel, who has let me share some of his expert knowledge about oscillators and phase locked loops, and Dr. Milos Krstic and Dipl.-Ing. Markus Ehrig, who both contributed to the implementation of the first 60 GHz baseband system. It was a great pleasure to work with them. I also have to thank Dr. Michael Methfessel for our fruitful technical discussions and his Latex support.

I would like to thank all my other colleagues at the Systems Department of IHP, who all contributed to a friendly working atmosphere and were always ready to provide help.

Last but not least, I would like to thank my family who has been a backbone during my whole life, in good and bad times, and were ready to support me whenever needed.

Maxim Piz

Contents

1 Introduction 1

2 Radio link model for 60 GHz transmission 6

2.1 Radio link based on direct conversion scheme . . . 7
2.2 Phase noise model for voltage controlled oscillator (VCO) . . . 8
2.3 Clock Jitter . . . 10
2.4 Analog-to-digital and digital-to-analog conversion . . . 12
2.5 Power amplifier nonlinearity . . . 16
2.6 I/Q mismatch . . . 20
2.7 Summarized link model, parameter investigation . . . 22

3 Channel models for 60 GHz radio communication 25

3.1 General characteristics of 60 GHz indoor channels . . . 25
3.2 HHI Channel model . . . 27
3.3 TG3c channel model . . . 29

4 OFDM under non ideal link conditions 34

4.1 OFDM modulation . . . 34
4.2 BPSK, QPSK and QAM-constellation mapping on subcarriers . . . 35
4.3 Degradation of OFDM due to imperfect synchronization and RF impairments . . . 36
  4.3.1 Residual intersymbol interference due to insufficient guard time length . . . 37
  4.3.2 Degradation due to frequency offset and phase noise . . . 41
  4.3.3 Degradation due to sampling clock frequency mismatch . . . 46
4.4 Carrier frequency offset estimation . . . 47
  4.4.1 Phase-noise induced frequency error . . . 49
  4.4.2 AWGN noise induced frequency error . . . 50
  4.4.3 Probability to exceed a given absolute frequency error . . . 52

5 PHY layer specification and performance investigation 54

5.1 Overview and general considerations . . . 54
5.2 Investigation of basic OFDM modulation parameters . . . 57
  5.2.1 Signal bandwidth . . . 58
  5.2.2 DFT size, guard length . . . 58
  5.2.3 Data, pilot and guard subcarriers . . . 59
  5.2.4 Pulse waveform and signal spectrum . . . 60
5.3 Selection of a channel code . . . 61
5.4 Stream arrangement and frame structure . . . 62
5.5 Interleaver design and convolutional code performance . . . 66
  5.5.1 Standard 802.11a-type interleaver . . . 66

- 5.5.2 Convolutional code performance using standard 802.11a-type interleaver . . . 67
- 5.5.3 "Folded" interleaver . . . 70
- 5.5.4 Performance comparison for standard and folded interleaver . . . 71
- 5.5.5 Interleaving scheme for WiMAX LDPC code (768,384), performance comparison 72

5.6 Preamble waveform design . . . 73
5.7 Preamble processing overview . . . 77
5.8 Frame detection, coarse timing synchronization and CFO estimation . . . 78
5.9 Channel estimation and equalization . . . 83
- 5.9.1 Channel estimation . . . 83
- 5.9.2 Equalization . . . 87

5.10 Fine time synchronization . . . 88
5.11 Tracking of phase and timing and channel re-estimation . . . 90
- 5.11.1 Tracking scheme for narrowband system (WIGWAM demonstrator) . . . 92
- 5.11.2 Improved tracking scheme for wideband system . . . 95

5.12 Receiver performance . . . 99
- 5.12.1 Performance of narrowband PHY for HHI channels . . . 99
- 5.12.2 Synchronization performance for wideband mode . . . 101
- 5.12.3 Performance of wideband PHY in static channel . . . 102
- 5.12.4 Performance of wideband PHY in residential time-variant NLOS channel . . . 104

6 Baseband processor implementation 107

6.1 Strategy for FPGA-based processor designs . . . 107
6.2 Transmitter . . . 107
6.3 Receiver overview . . . 108
6.4 Coarse synchronization and CFO estimation block . . . 113
- 6.4.1 Autocorrelator . . . 114
- 6.4.2 Antiphase detector . . . 115
- 6.4.3 Clustering logic, main controller and long autocorrelator . . . 116

6.5 Channel estimator and post-FFT timing estimator . . . 117
6.6 Pilot machine . . . 121
6.7 Data equalizer . . . 122
6.8 Four-port 256-point FFT . . . 123
6.9 Combined de-interleaving and depuncturing . . . 126
6.10 Viterbi decoder . . . 127
6.11 Digital automatic gain control . . . 132

7 Conclusion 139

A Derivation of the phase-noise induced CFO estimation performance 141

B Mathematical models for radio channels **144**
B.1 Continuous-time channel representation 144
B.2 Time-variant discrete-time channel representation 146
B.3 Static discrete-time channel model 146
B.4 Simplified time-variant discrete-time channel representation 147
B.5 Characterization of radio channels 148

C Mathematical basics of orthogonal frequency division multiplexing **151**
C.1 Continuous-time signal model 151
C.2 OFDM pulse waveform 153
C.3 Discrete-time signal model 154

D Convolutional codes and decoding algorithm **156**
D.1 Channel coding 156
D.2 Convolutional codes 158
D.3 Soft-decision bit metrics for convolutional codes 162

E BPSK, QPSK and QAM-mapping with gray-encoding **166**

F Partitioning algorithm for fast Fourier transform **167**

G OFDM PHY parameters for narrowband and wideband mode **169**

H Elementary hardware blocks **170**
H.1 CORDIC processors 170
H.2 Multipliers 173
H.3 $1/x$-Dividers 173
H.4 Delay elements 174

I Hardware platform **175**

References **175**

List of Figures **182**

List of Tables **185**

1 Introduction

Regardless of economical fluctuations, the demand for higher data rates in digital communication links has continuously grown over the last decade. This demand is not only related to data transmission via wired links, but also includes wireless services. Nowadays, wireless local area networks (WLANs) following the IEEE standards 802.11a, b, and g are an integral part of office networks. They also have reached a high degree of utilization in private households to connect computers to the Internet. The 802.11a standard operating at 5 GHz and 11g standard operating at 2.4 GHz both can provide a maximum theoretical data rate of 54 MBit/s. These standards make use of a modulation technique known as Orthogonal Frequency-Division Multiplexing (OFDM) in order to efficiently cope with multipath fading. More recently, the 802.11n standard has been finalized and promises much higher data rates by the use of broader channel bandwidths (40 compared to 20 MHz) and the MIMO (multiple-input, multiple-output) technique. 802.11n devices are equipped with multiple antennas and may provide a maximum theoretical data rate of 600 MBit/s using four spatial streams. In addition, the use of smart antenna technology offers a wider range. 802.11n, g, b standards and Bluetooth have to share the 2.4 GHz band. Whereas WLAN networks have to cover a typical range in the order of 10-100 meters, wireless personal area networks (WPANs) are intended for shorter range of about 1-10 meters. Bluetooth is a popular WPAN standard, which is mostly found in notebooks and mobile phones. Transmission is limited to 2.1 MBit/s. The so-called Ultra-Wideband (UWB) standard was planned as a major update in order to facilitate high-rate WPAN applications. This technique is based on the idea to spread the transmission power over a very large bandwidth of at least 500 MHz in order to avoid the violation of spectral masks defined by regulatory authorities. Unfortunately, two major consortiums involved in the standardization could not agree on a common standard, and a stalemate situation arose.

In retrospect, this happened in favor of a different breed of technology, namely *millimeter-wave wireless communication*, which had silently evolved when UWB was in the major focus of industrial interest. There are several strong arguments driving the development of millimeter-wave technology. Firstly, the FCC has reserved a bandwidth as high as 7 GHz for unlicensed short-range communication. This bandwidth is located at 57-64 GHz. Secondly, Silicon-Germanium and even CMOS technology have evolved that far as to provide the basis for fabrication of mid- to low-cost analog transceivers, avoiding expensive Gallium Arsenide solutions. In addition, these technologies allow the integration of the complete analog part. For these reasons, millimeter-wave communication has attracted more and more industrial power over the last years, and several new radio standards have developed or reached the state of finalization.

There are also some major drawbacks which can be associated with 60 GHz radio. Since millimeter waves experience very high attenuation when penetrating through walls, rooms act as natural cell boundaries for 60 GHz networks. Therefore, an access point is likely to be needed for every room where a 60 GHz communication link shall be established. This limits the variety of applications, where 60 GHz radio is reasonable. On the other hand, interference created by other cells sharing the same bandwidth is greatly reduced. Another severe drawback is the high free-space loss at 60 GHz. For high data-rate services using large amounts of bandwidth, the receiver sensitivity is significantly reduced, since the noise level in the low noise amplifier (LNA) is proportional to the utilized signal bandwidth. Therefore, for a typical transmitter output power, additional antenna gain is required to fulfill the link budget. Due to the small wavelength of 5 mm at 60 GHz which leads to a much smaller antenna form factor, this gain is much easier to achieve using either fixed-beam antennas or antenna arrays.

In 2005, the task group IEEE 802.15.3c was created for the development of a new 60 GHz radio standard for WPAN applications. This task group has finished their work in 2009 ([IEEb]). The new standard defines a single-carrier based mode as well as an OFDM-based mode. WirelessHD is a strong alliance of big companies in the field of consumer electronics ([Wir]). The aim is to establish a wireless HDTV link between a decoder and the TV set using *uncompressed* video transmission. This requires a data rate of at least 3 GBit/s. An overview of the WirelessHD standard, which uses OFDM, can be found in [Wir09]. In Europe, a 60 GHz standard was defined by ECMA in 2007. A more important association is the newer Wireless Gigabit Alliance (WiGig), which recently has developed their own standard for 60 GHz WLAN applications ([WiGa]). Key features of this standard have been adopted by the IEEE task group 802.11ad. This task group has the target to define a new 60 GHz WLAN standard.

Parallel to all these activities, a 60 GHz OFDM demonstration system has been independently developed by IHP Microelectronics within the German WIGWAM project ([WIGb]). This activity started when no 60 GHz radio standard existed and included the development of analog components, a PHY specification as well as the baseband processor implementation. This thesis focuses on the PHY specification as well as the baseband implementation of an OFDM system designed for 60 GHz radios. It covers the baseband specification and implementation work done for the IHP WIGWAM demonstrator, but also includes the follow-up project EASY-A ([Ena]), where much higher data rates have been envisaged. In summary, the suitability of OFDM modulation is investigated for 60 GHz indoor applications. This investigation includes a thorough performance analysis as well as implementation feasibility.

The thesis is organized as follows. Since the performance of the physical layer strongly depends on the analog front end characteristics, a 60 GHz radio link model is discussed in Chapter 2. This is followed by the characterization of two 60 GHz channel models in Chapter 3, which were used for system simulation. Chapter 4 deals with OFDM modulation under non ideal link conditions. Except IQ-mismatch, all important causes of performance degradation are highlighted and the degradation is quantified. Furthermore, carrier frequency offset correction in OFDM systems is reconsidered for the given link conditions. These sections can be regarded as a foundation for Chapter 5. In this section, the whole PHY layer for both systems together with receiver algorithms is developed. OFDM PHY parameters for optimized performance are investigated. Algorithms for synchronization, channel estimation and tracking are elaborated. The performance of different coding schemes and interleaving patterns is studied. At the end of the chapter, the receiver performance under realistic link conditions is investigated. Chapter 6 deals with the challenging implementation of the system on an FPGA platform limited in clock speed. A high degree of parallelization is required to process the large bandwidth and data rate. In summary, despite the experienced high latency and low clock speed, it is shown that the system is feasible and can achieve good performance under the given design restrictions. In order to facilitate readability of this work, the required basics of modulation and coding were put into the Appendix. This also includes long calculations. All system simulations were done in MATLAB and programmed from scratch. The implementation of hardware components was done in VHDL.

The PHY used for the WIGWAM demonstrator will be called the *narrowband PHY*. Accordingly, the PHY for the EASY-A demonstrator will be referred to as the *wideband PHY*. This second PHY is based on the narrowband, but offers improved features and algorithms.

List of abbreviations

ACF	autocorrelation function
ADC	analog-to-digital converter
AGC	automatic gain control (loop)
AWGN	additive white Gaussian noise
BER	bit error rate
BPSK	binary phase shift keying
CCF	cross-correlation function
CDF	cumulative distribution function
CFO	carrier frequency offset
CORDIC	COordinate Rotation DIgital Computer
CP	cyclic prefix
CPE	common phase error
CSI	channel-state information
DAC	digital-to-analog converter
dBu	dB "unity"
DFT	discrete Fourier transform
FER	frame error rate
FFT	fast Fourier transform
FIFO	first-in first-out standard memory block
FPGA	field-programmable gate array
FSM	finite-state machine
IBO	input power backoff
ICI	intercarrier interference
ISI	intersymbol interference
LDPC	low-density parity-check code
LNA	low noise amplifier
LOS	line-of-sight
NLOS	non-line-of-sight
OBO	output power backoff
OFDM	orthogonal frequency division multiplex
PDP	Power delay profile
PHY	Physical Layer
QAM	quadrature amplitude modulation
QPSK	quadrature phase shift keying
RMS	root-mean square
RS	Reed-Solomon (code)
SF	signal field

SIR	signal-to-interference ratio
SV	(S)aleh-(V)alenzuela (channel model)
TDMA	time-division multiple access
TSV	(T)wo-path (S)aleh-(V)alenzuela (channel model)
VCO	voltage-controlled oscillator

2 Radio link model for 60 GHz transmission

The basic task of a digital radio system is to reliably transport data from some station A to some station B via wireless radio transmission. For this purpose, digital data is transformed in some way into an analog time-continuous signal, which is amplified and radiated through the antenna. The radio signal is designed to be bandlimited and uses some preallocated part of the radio spectrum, a dedicated *frequency channel* within a (broader) service band. The transmitted radio signal wave will propagate to the receiver using different paths and will arrive with different delays, amplitudes and phases at the receiver antenna. As a result of this, the "channel" between the radio devices acts like as a linear system, defined by its impulse response. The receiver must be able to restore the message even after this linear distortion for most of the cases. The channel may be time-variant due to the motion of the radio devices or other obstacles. The situation is further complicated by the impact of non-ideal components of the analog front end. These components have limitations, which also impair the performance of the system. Due to the random nature, the radio channel can only be characterized by statistical properties, which depend on many parameters. These are influenced by the application scenario, i.e. the environment where communication takes place and the distance of the stations. The channel characteristics also depend on the location of the service band and the signal bandwidth. Finally, the antenna radiation pattern is able to strongly influence the channel.

The transmission scheme should be designed as to deliver good link performance for the observed channel characteristics and expected properties of the analog front end. Since the received signal strength diminishes with increased distance, performance is usually measured against a range of *signal-to-noise ratios* (SNR) at the receiver to get range-independent results. Performance may be characterized in terms of SNR-dependent *bit error rate* (BER) or *packet error rate* (PER), assuming burst data transmission. From such a curve, a minimum SNR value can be extracted for which either BER or PER do not exceed a given value. From this minimum SNR value, one may derive a *link budget* and estimate an average maximum distance of reliable operation. The BER- or PER-performance are a criterion of *power efficiency*. Another criterion is the *bandwidth- or spectral efficiency*, which is given as the ratio of useful data rate versus occupied bandwidth in bit/s/Hz. Higher power or spectral efficiency may come at the price of increased system complexity or put a higher demand on analog front end components.

From the mathematical viewpoint, the transmitted radio signal represents a bandlimited bandpass signal $f(t)$ at some center frequency f_c. It can be described by inphase- and quadrature-components $I(t)$ and $Q(t)$, which constitute baseband-signals with low-pass character ([Joh01]). The relationship is given by

$$f(t) = I(t) \cdot \cos(2\pi f_c t) - Q(t) \cdot \sin(2\pi f_c t) = Re\{(I(t) + j \cdot Q(t)) \exp(j 2\pi f_c t)\} \tag{1}$$

For system simulation, a realistic link model is needed, which comprises the radio channel and the impairments of the analog front ends. To limit simulation time, it is advantageous to avoid any translation to some carrier frequency and back. Instead, the radio link is usually modeled entirely in the baseband

and operations are applied on the I/Q components.

2.1 Radio link based on direct conversion scheme

The *direct-conversion scheme* for the transmitter is a technical realization for the frequency shift performed with Equation (1) and can act as a reference model for this work. It embraces the main essential impairments of the analog front end, which contribute to a loss of performance. Figure 1 illustrates the transmission model for a radio link based on the direct-conversion scheme for both sides.

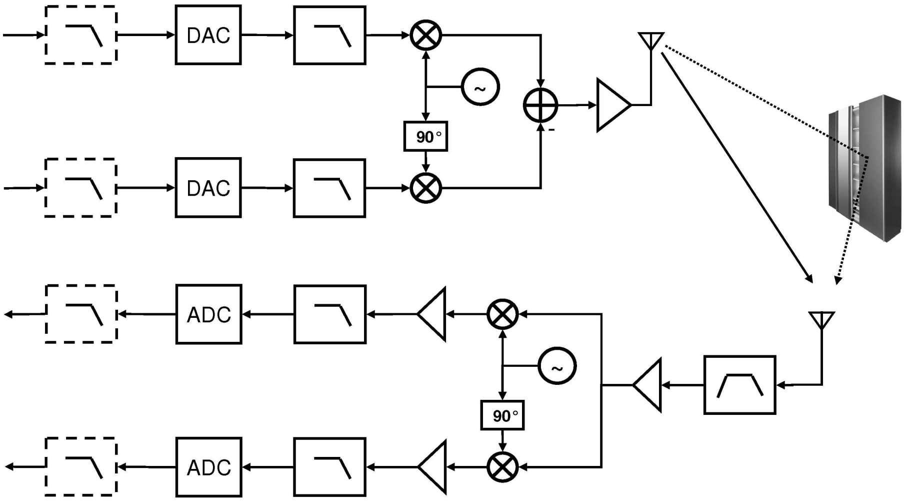


Figure 1: Transceiver model

After digital-to-analog conversion, the inphase and quadrature signals pass an analog reconstruction filter to remove out-of-band images. Then the signal is upconverted to radio frequency with a quadrature mixer. According to (1), two sinusoidal signals with 90 degree phase shift are needed. They are often generated with a voltage-controlled oscillator and a 90 degree phase shifter. The synthesized RF signal is amplified by the power amplifier (PA) and radiated through the antenna. The channel causes multipath propagation and acts in this way as a linear time-variant or invariant filter. If time-variant, the channel may introduce Doppler shift if the direct path strongly dominates. Otherwise, in case of significant multipath reception, Doppler spread appears. Due to free space loss, the radio signal arrives with a level which is orders of magnitude lower than transmitted. After passing a broader bandpass filter to reject interference out of the service band, some amplification is applied to the signal by a low-noise amplifier (LNA). The signal is then downconverted to baseband. Channel selection is done by choosing the same carrier frequency at the receiver side. Low-pass filters in the baseband filter out all other unwanted signals in the neighborhood channels. Additional variable gain amplification is applied to bring the signal to a defined level suitable for A/D-conversion. The low-pass filters act also as anti-alias filters for this purpose.

In an analog receiver front end, the signal level will usually rise from stage to stage. The first few stages define the sensitivity of the receiver, since the signal level is low and picked up noise at these stages has the highest contribution in terms of lowering the SNR ([Beh98]). The linearity of the receiver will usually be determined by the stages where signal level is highest. For the transmitter, the signal will have a level high enough at all stages to avoid signal degradation through noise. On the other hand, the power amplifier is the most critical device in terms of linearity. For high signal levels the PA may start to compress the signal and produce harmonics and intermodulation products. Another type of is *phase noise* of the oscillators used for frequency translation. Multicarrier modulation is particularly sensitive to phase noise perturbation. In addition, phase noise may show up through timing jitter of the clock used for A/D- and D/A-conversion. Yet another sort of impairment consists of mismatch between the inphase and quadrature-path (*I/Q mismatch*), which causes a mirroring of positive and negative frequencies. Depending on the origin, I/Q mismatch might be frequency dependent or independent. In the following, we consider phase noise, clock jitter, A/D-D/A conversion, PA nonlinearity and I/Q mismatch. Channel models are discussed in Section 3.

Actually, the 60 GHz transceiver, which was employed during the project, has a *super-heterodyne* architecture. For this type of transceiver, the problem of the *image band* appears, which must be sufficiently attenuated. This problem is of analog nature and can be resolved by proper circuit design. If sufficient image rejection is maintained, both types of transceivers exhibit the same mechanisms of signal deterioration, namely I/Q mismatch, amplifier nonlinearities, additive noise, phase noise and DC-offset, but just to a different degree. In general, the direct conversion scheme suffers from higher DC offset and I/Q mismatch, but we can take it just as a generic model and virtually replace the superheterodyne transceiver with an almost equivalent direct-conversion model having nearly the same parameters. This can be done if the IF stage causes little degradation or if all effects at the IF stage can be incorporated into equivalent noise sources at the RF stage. Indeed, the RF power amplifier in the transmitter usually causes the highest nonlinearity, the noise floor is largely determined by the receiver low-noise amplifier at RF and the phase noise is dominated by the VCOs with the highest frequency, which are employed at the RF stage.

2.2 Phase noise model for voltage controlled oscillator (VCO)

In a typical radio system, at least one reference oscillator is needed for the frequency translation of the signal in the RF front end and as a clock source for the A/D-D/A converters and the digital baseband. Oscillator phase noise introduces clock jitter in converters and also deteriorates the signal by phase modulation of the carrier. In any phase noise model the oscillator signal is described via

$$x_{vco}(t) = A \cdot cos(\omega_0 t + \phi(t)) \tag{2}$$

where $\phi(t)$ is the random phase associated with phase noise. A widely used model for phase noise system analysis is the *free running oscillator* based on diffusion theory ([F. 04], [T. 95], [Luc98]) also used to describe *Brownian motion* of particles in Physics. In this model, the phase perturbation is created according to the Wiener process given by

$$\phi(t) = \int_{\tau=0}^{t} \Delta\omega(\tau) d\tau \tag{3}$$

$$S_{\Delta\omega}(\omega) = 2D_{\phi}, \quad R_{\Delta\omega\Delta\omega}(\Delta t = t - \tau) = \langle \Delta\omega(t)\Delta\omega(\tau) \rangle = 2D_{\phi}\delta(t - \tau) \tag{4}$$

where $\langle ... \rangle$ in (4) denotes expectation. The phase perturbation $\phi(t)$ is obtained via integration of an instantaneous frequency deviation $\Delta\omega(t)$, which itself constitutes white Gaussian noise ([F. 04]) with constant two-sided power spectral density $S_{\Delta\omega}(\omega)$ and autocorrelation $R_{\Delta\omega\Delta\omega}(\Delta t)$. The disturbing phase (or excess phase) $\phi(t)$ is assumed to be zero at t=0 and diffuses away with increasing time. D_{ϕ} is called the *diffusivity*. From (2) to (4), the mean, autocorrelation function (ACF) and variance can be derived. The mean of $\phi(t)$ is zero, but the variance increases linearly with time. This reflects that the Wiener process is non-stationary.

$$R_{\phi\phi}(t, \tau) = 2 \cdot D \cdot min(t, \tau) \quad \sigma^2_{\phi}(t) = R_{\phi\phi}(t, t) = 2D_{\phi}t \quad E\{\phi(t)\} = 0 \tag{5}$$

Due to the integration, the power spectral density (PSD) of the phase is equal to

$$S_{\phi}(w) = S_{\Delta\omega}/\omega^2 = 2D_{\phi}/\omega^2 \tag{6}$$

The carrier signal $x_{vco}(t)$ becomes stationary if one adds a constant phase shift randomly distributed over $[0, 2\pi]$ and seen at t=0. The ACF of the carrier signal $x(t)$ can also be calculated from (2) to (4). From the ACF, the single sideband power spectrum, which is normalized to the carrier power, is obtained using the Wiener-Khinchin theorem:

$$S_{SSB}(\omega) = \frac{2D_{\phi}}{(\omega - \omega_0)^2 + D^2_{\phi}} \tag{7}$$

At the carrier frequency, this PSD approaches a plateau with a two-sided 3-dB linewidth of $\Delta f_{\text{3dB}} = 2D_{\phi}/(2\pi) = D_{\phi}/\pi$. At sufficient distance from the carrier, the phase noise decreases with a slope of -20 dB/decade and coincides with the PSD of the phase noise $\phi(t)$.

$$|\omega - \omega_0| \gg D_{\phi} \Rightarrow S_{SSB}(\omega - \omega_0) \approx S_{\phi}(\omega) \tag{8}$$

Computer simulations can be performed using a discrete-time model for some sampling frequency f_T. From (3) we get for $t = nT, T = 1/f_T$

$$\phi(t) = \int_{\tau=0}^{t} \Delta\omega(\tau)d\tau = \sum_{k=1}^{n} \int_{\tau=(k-1)T}^{kT} \Delta\omega(\tau)d\tau = \sum_{k=1}^{n} \Delta\Omega_k \tag{9}$$

The integral is split-up into n integrals $\Delta\Omega_k := \int_{\tau=(k-1)T}^{kT} \Delta\omega(\tau)d\tau$ of duration T. Since $\Delta\omega(t)$ is Gaussian-distributed, the same applies to Ω_k (and to $\phi(t)$). Hence to obtain $\phi(nT)$, we simply sum-up n random numbers taken from the normal distribution with variance $\sigma^2_{\Delta\Omega} = 2D_{\phi}T$. Sampling the phase deviation $\phi(t)$ in this way leads to aliasing errors, since $\phi(t)$ has infinite bandwidth. Hence the power spectrum of the discrete phase within the simulation bandwidth is slightly higher than the true spectrum. Since the noise contributions far from the carrier (assumed in the vicinity of f=0 Hz) are very small, this effect is normally negligible.

For PLL based synthesizers, the phase noise spectrum is composed of high-pass filtered VCO noise spectrum and low-pass filtered (and amplified) noise spectrum of the reference oscillator ([F. 05]). From a system perspective, given some conventional digital modulation technique, one may call a PLL as being *narrowband*, if the loop bandwidth is considerably lower than the inverse of the *symbol time*. In this case, the carrier noise spectrum is dominated by the noise contribution of the VCO within the frequency band of interest. As an example, in OFDM systems, the *common phase error correction* in the receiver will compensate for the low frequency content of the carrier phase noise ([F. 05]) and therefore greatly reduce reference phase noise. Hence the Wiener model can be applied not only for free running oscillators but also as an approximation for narrowband PLLs.

In practice, the single sideband phase noise $L_{\mathrm{ssb}}(\Delta f)$ of some oscillator is measured at a frequency offset Δf Hz from the carrier frequency, within the -20 dB/decade region. With the measured single side-band phase noise value, the diffusivity is calculated using (6). In a radio system, both transmitter (TX) and receiver (RX) VCO contribute to the total amount of phase noise. To simplify simulation, one may merge both contributions into one fictive equivalent noisy receiver VCO with diffusivity

$$\widetilde{D}_{phi} = D_{tx} + D_{rx} \tag{10}$$

leaving a virtually noiseless transmitter VCO. This greatly simplifies analysis of other tasks like *channel estimation performance* as will be described in later chapters.

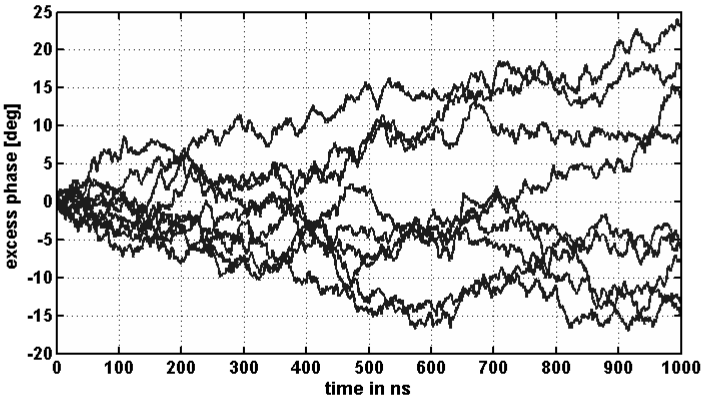


Figure 2: Different realizations for the random excess phase $\phi(t)$ according to Wiener model

in Figure 2, a series of different realizations of the random phase $\phi(t)$ is depicted for a oscillator having a single-sideband phase noise spectrum of L_{ssb} = -90 dBc/Hz at 1 MHz offset. Over just a microsecond, the phase may deviate as much as 25 degree. This shown noise performance is typical for an integrated oscillator using Si-Ge technology. Therefore, it is for example nearly impossible to transmit an OFDM signal according to the WLAN standard 802.11a with an OFDM symbol rate of 4 μs with a 60 GHz radio front end featuring this oscillator.

2.3 Clock Jitter

A VCO is also used as a clock source for the data-converters and the digital system. Due to the much lower frequency, the oscillator can be realized with higher purity. On the other hand, radio signals with very high bandwidth are more susceptible to clock jitter effects. Therefore the impact of clock uncertainty plays a crucial role for 60 GHz designs. The sampling instants may be defined as the zero crossings of a sinusoidal signal in positive direction. We assume that the receiver clock is *not synchronized* to the transmitter VCO, since transmitter and receiver baseband are running with independent clocks and clock recovery is done in the digital domain. We further assume that the VCO is a free running oscillator, where this time the cosine is replaced with a sine function for convenience.

$$x_{vco}(t) = A \cdot sin(\beta(t)) = A \cdot sin(\omega_T t + \phi(t)) \tag{11}$$

$\omega_T = 2\pi f_T$, f_T being the ideal sampling frequency. The sampling instants could be defined as the zero crossings of the sinusoidal in positive direction. It turns out that such a definition is misleading, because the random phase contribution $\phi(t)$ may provoke two or even more zero crossings anytime when the oscillator phase $\beta(t)$ is in the vicinity of 0 radians. Instead an approximation is made to arrive at a time series t_n of the jittering sample instants. The n-th sample instant t_n can be written as an accumulation of n elapsed clock cycles:

$$t_n = \sum_{k=1}^{n} T_k, \quad t_0 = 0 \tag{12}$$

The n-th clock cycle duration T_n is approximately equal to

$$T_n = T + \Delta T_n \approx T + \Delta\phi(n)/\omega_T \tag{13}$$

where $\Delta\phi(n)$ is a random phase change of the VCO caused by phase noise, which appears within the *ideal* interval T and leads to a timing shift. For the Wiener process, this phase change is obtained by integration of white noise over T seconds. Combining (12) and (13) yields

$$t_n = nT + \Delta t_n \approx nT + (1/\omega_T) \sum_{k=1}^{n} \Delta\phi(k) \tag{14}$$

But $\Delta\phi(k) \equiv \Delta\Omega_k$ in (9), and it follows from (9) that (14) can be written as

$$t_n = nT + \Delta t_n \approx nT + \phi(nT)/\omega_T \tag{15}$$

Comparing (9) and (14) reveals that according to this approximation, the accumulated time shift Δt_n is directly obtained from the diffusing phase $\phi(t)$ at time instant nT. If a noise-free sinusoidal signal $x(t) = A \cdot \sin(2\pi f_c t)$ is sampled with the noisy clock signal, the resulting sampled signal reads

$$x_n := x(t_n) = A \cdot \sin(2\pi (f_c/f_T) \cdot n + (f_c/f_T) \cdot \phi(nT)) \tag{16}$$

Therefore, sampling a noise-free sinusoid with a free running oscillator with excess phase $\phi(t)$ is formally equal to sampling a sinusoid with excess phase $\tilde{\phi}(t) = (f_c/f_T)\phi(t)$ with an ideal clock. In effect, the phase noise spectrum of the noisy clock is projected onto the sampled sinusoidal and the power spectrum is amplified by a factor of $(f_c/f_T)^2$. In an OFDM system this leads to intercarrier interference, which rises for higher frequency and may degrade the effective SNR of outer subcarriers.

The *cycle jitter* ΔT is defined as the rms value of the deviation from the ideal cycle duration T ([Fra99]). Hence from (13) we have

$$\Delta T = \sqrt{E\left\{{\Delta T_n}^2\right\}} = \sqrt{E\left\{(\Delta\phi(n)/\omega_T)^2\right\}} = \sqrt{2D_\phi T/{\omega_T}^2} = \sqrt{2D_\phi T^3}/2\pi \tag{17}$$

The rms value for the accumulated timing error after n cycles can be expressed as

$$\sqrt{E\left\{\Delta t(n)^2\right\}} = \sqrt{E\left\{(\sum_{k=1}^{n}\Delta T_k)^2\right\}} = \sqrt{\sum_{k=1}^{n} E\left\{(\Delta T_k)^2\right\}} = \sqrt{n}\cdot\Delta T \tag{18}$$

because the partial timing errors are statistically independent. Since the accumulated timing error is proportional to the excess phase of the free running VCO, it is a Gaussian random variable with zero mean and variance

$$\sigma^2_{\Delta t}(n) = n\cdot\Delta T^2 \tag{19}$$

2.4 Analog-to-digital and digital-to-analog conversion

Assuming a realistic spectral efficiency of no more than 4 Bit/s/Hz in a mobile, frequency selective environment, signal bandwidths of several hundred MHz are required to achieve a source data rate of around one Gigabit/s for a single-input single-output (SISO) radio link. According to the sampling theorem, AD/DA conversion with a sampling rate f_T of at least the RF signal bandwidth is needed. The required devices still represent the leading edge in current development and not many companies offer converters with sampling rates beyond one Gigasamples/s. Both the performance and the cost are critical factors and realistic converter models should be included in the simulation. We start from an ideal converter model and move later to a more realistic one. The A/D converter is considered at first. The ideal N-bit midread-quantizer for an amplitude range $[-A, A)$ and step size Δ follows the rule

$$Q(x) = [min(max(x, x_{min}), x_{max})/\Delta]\cdot\Delta \tag{20}$$

$$x_{min} = -A \quad x_{max} = A(1-1/2^N) \quad \Delta = A/2^N$$

where [...] in the above equation (20) denotes rounding. The min-max-operation limits the input signal to the range of the converter and signals outside this range are clipped. It is well known that the ideal quantization noise, under the idealized assumption of no correlation with the signal, has a power of $P_Q = \Delta^2/12$. For a sinusoidal input with full amplitude A and a power level of $A^2/2$, the ratio between signal level and quantization noise will be $\mathrm{SNR}_{\mathrm{sin,dB}} = 10\log_{10}((3/2)\cdot 2^{2N}) = 1.76 + 6.02\cdot N$ [dB]. In the more general case, an arbitrary input signal will have a power level P_{in} and the SNR will be

$$\mathrm{SNR}_{\mathrm{dB}} = 4.76 + P_{\mathrm{sig,dBFS}} + 6.02\cdot N\ [\mathrm{dB}] \tag{21}$$

For the equation to hold, the converter must be ideal and the input signal must not clip (or very rarely). The input power level on the right hand side is given in relation to the full scale power level A^2 in dBFS (dB-full-scale). For example, an OFDM input signal, which is approximately Gaussian, would typically drive the converter at an input level of around P_{sig} = -18 dBFS in order to prevent clipping most of the time and also to retain enough resolution. In this case, there are 3 bit of headroom between the root-mean-square (rms) value and full-scale. Due to device imperfections, the SNR performance of *real* converter will be somewhat below the ideal value, therefore the resolution in terms of *effective number of bits* (ENOB) will not reach the nominal value. The ENOB is calculated from the measured SNR for a sinusoidal input via

$$ENOB_{dB} = (SNR_{dB} - 1.76)/6.02 \qquad (22)$$

The bit degradation $N_{\text{Bit}} - ENOB$ can originate from various detrimental effects including thermal noise, clock and aperture jitter and nonlinearity. In addition, ENOB is in general frequency dependent and declines with rising input frequency if clock jitter is present.

ADCs can be characterized by quasi-static (integral non-linearity, differential non-linearity) and dynamic performance [[R.H99]]. For applications in communications, dynamic performance is usually most important. Measurements include the signal-to-noise ratio (SNR) and the *spurious free dynamic range* (SFDR).

The SFDR is a way to measure the *linearity* of the converter. The converter does not exactly replicate the ideal clipping curve given in (20). The SFDR is the difference between the input full-scale sinusoidal and the largest spur in the spectrum. SFDR is an important measure for multi-channel applications like *software radios*, where many narrowband channels are received at once. In a software radio, the receiver may sample a complete service band consisting of many narrowband frequency channels from different stations spread across the spectrum. The different channels may differ in power by several orders of magnitude. To prevent such effects like desensitization, blocking, cross- and intermodulation ([Beh98]), the converter must have a very high dynamic range and very good linearity. For conventional single-channel systems on the other hand, dynamic range and linearity requirements are relaxed. Adjacent channels can be filtered in the RF front end and an automatic gain control (AGC) stage in front of the converter ensures a proper input signal level. *It is therefore assumed for this application that the converter always satisfies the linearity requirements leaving the SNR as the main performance parameter*.

[[R.H99]] represents a good survey although not the newest report, which has highlighted the limitations of A/D converter technology. In general, higher resolutions are achieved with lower sampling frequencies. In addition to quantization noise, other mechanisms exist to lower performance. For high-resolution converters with low to moderate speed below some MSPS, thermal noise, arising from the analog circuitry, is the limiting factor. For high speed ADCs, up to several GBit/s, *aperture jitter* takes over and puts the bound on attainable resolution. For even higher sampling rates, *comparator ambiguity* takes over as a third bound. *It shall be assumed in this work that the required converters fall in group 2, meaning that aperture jitter is the main cause for a performance degradation in the converter.*

Aperture jitter is a measure for the variation of the sampling instants. It is often defined as the standard deviation from the ideal sampling point. For real sample instants t_n and ideal sample instants

$t_{n,ideal}$, this jitter is defined for the timing deviation $\Delta\tau(n)$ as

$$\tau_{\text{rms}} = \sqrt{E\{\Delta\tau_n\}^2} = \sqrt{E\left\{(t_n - t_{n,ideal})^2\right\}} \tag{23}$$

This definition can only be applied for a time deviation process $\Delta\tau(n)$, which is stationary with zero mean. To measure this jitter in time domain, synchronization of the input signal and the clock must be performed. Now the question arises how this jitter relates to phase noise induced jitter discussed in the last section. We note that clock deviations can also be caused by internal converter circuitry. Then aperture jitter can arise from both oscillator phase noise, which may undergo some form of high-pass filtering during synchronization, and by circuit noise. We consider a sinusoidal input signal $v_s(t) = A \cdot \sin(2\pi f t)$. Note that if the circuit induced jitter has a white spectrum, the noise floor of the output spectrum is simply increased, whereas phase noise induced jitter will produce narrowband noise around the carrier, as discussed in the last section. Unfortunately, the data converter vendor does not provide enough information to identify the dominant noise source. *We will assume that measurements have been obtained with a high purity clock and that the jitter given in (23) solely originates from circuit noise, which is modeled as AWGN.* We can define a jitter model which includes this white component $\Delta\tau_n$ and a non-stationary component Δt_n, as defined in (15). The sampling instants are written as

$$t_n = nT + \Delta\tau_n + \Delta t_n \tag{24}$$

Consider for the moment only AWGN jitter with no clock jitter ($\Delta t_n = 0$). Since this jitter is very small, the amplitude error $\Delta v(n)$ caused by an timing error of $\Delta\tau_n$ for a sample at time nT is approximately

$$\Delta v(n) = \Delta\tau_n \cdot (\frac{dv}{dt})_{t=nT} \tag{25}$$

The rms error for an arbitrary input signal can be calculated using

$$\Delta v_{rms} = \sqrt{\langle \Delta v^2 \rangle} = \sqrt{\left\langle (\frac{dv}{dt} \cdot \Delta t)^2 \right\rangle} = \sqrt{\left\langle (\frac{dv}{dt})^2 \right\rangle} \cdot \tau_{\text{rms}} \tag{26}$$

For a input sinusoidal with frequency $f < f_T/2$ we obtain

$$\Delta v_{s,rms} = A\pi f \tau_{\text{rms}} \tag{27}$$

Hence, the error variance $(\Delta v_{s,rms})^2$ also increases with the square of the frequency as did the phase noise spectrum in (16). We can further extend the model for the AD-converter by adding a AWGN noise source at the input. This noise contribution is attributed to thermal noise. This may be necessary if a resolution loss is observed at very low frequencies, i.e. if the ENOB in Equation (22) falls below the bit resolution.

Let's assume an OFDM signal at the input and approximate it as ideal bandlimited white noise $v_n(t)$ with one-sided bandwidth B and noise power $\sigma^2_{v_n}$ [1]. With the aid of the rectangular function defined as

$$\begin{aligned} \Pi(x) &= 1 \quad \text{for} \quad -1/2 \le x < 1/2 \\ \Pi(x) &= 0 \quad \text{else} \end{aligned} \tag{28}$$

---

[1] there is no spectral content for frequencies below -B or above +B

the spectral density of $v_n(t)$ is $S_{v_n}(f) = \sigma_{v_n}^2/(2B) \cdot \Pi(f/(2B))$. Hence the spectral density of the derivative equals $S_{dv_n/dt}(f) = \sigma_{v_n}^2/(2B) \cdot \Pi(f/(2B)) \cdot (2\pi f)^2$. If we assume only white jitter noise, we obtain an error power of

$$\begin{aligned}(\Delta v_{n,rms})^2 &= \tau_{\text{rms}}^2 \cdot \left\langle (\frac{dv}{dt})^2 \right\rangle = \tau_{\text{rms}}^2 \cdot \int_{f=-\infty}^{+\infty} S_{dv_n/dt}(f)df \\ &= \frac{(2\pi\tau_{\text{rms}}\sigma_{v_n})^2}{2B} \int_{f=-B}^{+B} f^2 df = \frac{(2\pi B\tau_{\text{rms}}\sigma_{v_n})^2}{3}\end{aligned} \tag{29}$$

We summarize that the SNR for a sinusoid with frequency f and the SNR for an OFDM signal with I/Q bandwidth B (corresponding to bandwidth $2B$ at radio frequency) are bounded by very similar values:

$$\widehat{SNR}_{\text{dB,sin}} = 20 \cdot \log_{10}(A/(A\pi f\tau_{\text{rms}})) = -20 \cdot \log_{10}(\pi f\tau_{\text{rms}}) \tag{30}$$

$$\widehat{SNR}_{\text{dB,noise}} = 10 \cdot \log_{10}(3/(2\pi B\tau_{rms})^2) = -20 \cdot \log_{10}(\pi B\tau_{\text{rms}}) - 1.25\text{dB} \tag{31}$$

Until now, we have focused on the AD-converter. For this component, ideal sampling as the process of picking an instant input value is achieved with the sample-and-hold circuit. Now we consider the DA-converter. According to the sampling theorem, the analog waveform $z_A(t)$ must be reconstructed from a discrete time sequence

$$z(n) = x(n) + j \cdot y(n) \tag{32}$$

by the well known interpolation formula

$$z_A(t) = \sum_{n=-\infty}^{+\infty} z(n) \frac{\sin(\pi(\frac{t}{T} - n))}{\pi(\frac{t}{T} - n)} \qquad T = 1/f_T \tag{33}$$

But the output of a technically feasible digital-to-analog converter will usually consist of a step function of the time-discrete input according to

$$z_S(t) = \sum_{n=-\infty}^{+\infty} z(n)\Pi(\frac{t-nT}{T}) \tag{34}$$

With no sampling jitter, this corresponds to a weighted Dirac comb passing a sample-hold device with impulse response and transfer function

$$h(t) = \Pi(t/T) \quad H(f) = T \cdot Sinc(fT) \tag{35}$$

where the Sinc function is defined as

$$Sinc(x) = \frac{\sin(\pi x)}{\pi x} \tag{36}$$

The transfer function of the impulse shapes the output spectrum and attenuates the upper frequencies within the fundamental band. This effect can be easily compensated in the digital domain. The output step signal $z_S(t)$ must pass an analog reconstruction filter. A difficulty arises if we include the jitter effects. We will observe pulse width variations of the output signal $z_S(t)$ and the simple linear relationship in (34) is not valid any more. We may rather avoid the exact modeling of this behavior because it requires a very large oversampling rate already *prior* to the reconstruction filter. Two approximations seem possible: The first method consists in creation of the ideal transmit I/Q signal and immediate resampling using a jittering clock. Alternatively, since the jitter is small, the transmitter clock, clean or noisy, can be simply declared as the reference clock so that the receiver clock is set in relation to it. Then all jitter noise contributions are unified in one AD-converter model, similar to the approach made for phase noise where both contributions are modeled with the receiver VCO (Equation (10)). These sources of degradation are used in the link model shown in Figure 5 in Section 2.7.

2.5 Power amplifier nonlinearity

The allowed driving of a power amplifier (PA) in the transmitter is an important factor in system design. If the modulation scheme allows heavy saturation of the PA without causing self-interference, the PA can operate with high efficiency. This is the case for constant-envelope modulation schemes ([Beh98]) where information is entirely mapped on the carrier phase or frequency. In this way, power is saved in the transmitter. Harmonics are suppressed by filters and the frequency characteristic of the antenna.

Modulation schemes which do not feature a constant envelope, usually require an operation in the linear region of the PA. Especially schemes with a high *peak-to-average power ratio* (PAPR) demand for high linearity, which is managed with a high *power back-off*. The output backoff (OBO) is defined as the ratio of saturation power P_{sat}, which is the highest output power that can be delivered by the particular amplifier, to the average output power of the signal.

Unfortunately OFDM belongs to the schemes with high PAPR. The theoretical PAPR of the signal waveform created from N subcarriers is equal to N ([Ric00]). Hence, a extremely large back-off would be required to avoid any hypothetical overdrive. In practice, the back-off is chosen such that overs can be tolerated in the system. The amplitude of an OFDM signal with a large amount of subcarriers is nearly Rayleigh-distributed, and high overs occur only rarely.

A survey reveals that the saturation power of a state-of-the-art 60 GHz Si-Ge amplifier is below or in the order of the allowed transmit power. This means that the chosen back-off for a typical 60 GHz PA is paid with a reduction of potential transmission power, which is unused. On the other hand, continuously lowering the output back-off leads to some point where saturation will dominate the bit error rate. In summary, an optimum may be found which maximizes the link budget.

As an early example, the optimum back-off value for uncoded OFDM with N=512 and N=1024 subcarriers was investigated for 4-DPSK in a flat channel in [Chr91]. An OBO of about 5.3 dB was obtained as the optimum. This work had also suggested a mathematical model for a solid-state amplifier, which has been widely used in the research area. The *Rapp model* was also chosen as the reference model for system comparison in extended form in the IEEE task group 802.15.3c ([IEE07]).

For this model, the PA is assumed to be *memoryless*, and the nonlinear effects depend solely on the input amplitude. For this type of model, a sufficiently high input amplitude A causes a distortion of output amplitude and/or phase, named AM/AM and AM/PM conversion. If the input waveform at carrier frequency f_c reads

$$s_{in} = A(t)cos(2\pi f_c t + \phi(t)), \quad A(t) \geq 0 \tag{37}$$

the PA output signal can be written as

$$s_{out} = g(A(t))cos(2\pi f_c t + \phi(t) + \Phi(A(t))) \tag{38}$$

where $g(A)$ and $\Phi(A)$ are the AM/AM and AM/PM conversion functions. (38) does not include created harmonics of the carrier as multiples of the carrier frequency, since these are assumed to be sufficiently attenuated. The complex baseband envelopes of input and output signal are written as

$$\begin{aligned} \widehat{s}_{in} &= A(t) \cdot e^{j\phi(t)} \\ \widehat{s}_{out} &= g(A(t)) \cdot e^{j\phi(t)+j\Phi(t)} \end{aligned} \tag{39}$$

According to the Rapp model, the AM/AM conversion function is given as

$$g(A) = \frac{v \cdot A}{(1 + (\frac{v \cdot A}{A_0})^{2p})^{\frac{1}{2p}}} \Leftrightarrow g(A)/A_0 = \frac{A/A_0'}{(1 + (\frac{A}{A_0'})^{2p})^{\frac{1}{2p}}}, \quad A_0' = A_0/v \tag{40}$$

A_0 specifies the limiting output amplitude, v is the small signal gain and the free parameter p sets the smoothness of the transition from linear to the limiting region and will be called the *knee factor*. p must be matched to the considered amplifier device. For $p \to \infty$, the curve approaches an ideal limiter. Furthermore, Rapp assumes that the AM/PM conversion can be neglected for the solid-state PA.

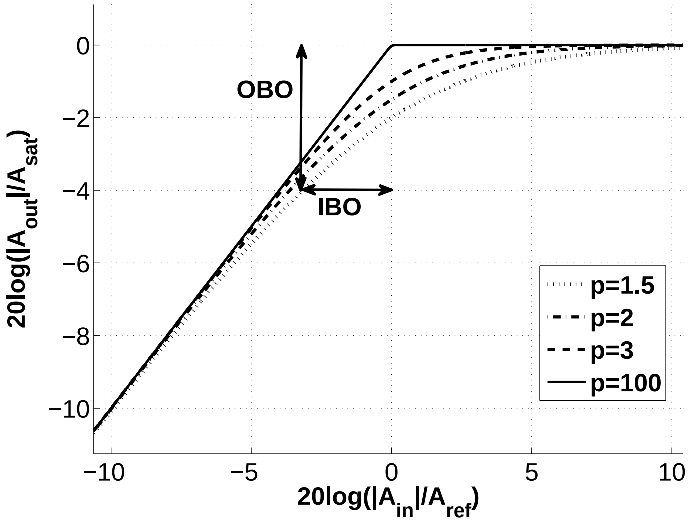


Figure 3: AM/AM-conversion according to Rapp's model

On the right side of (40), the substitution $A_0' = A_0/v$ has been used. A_0' is the instantaneous input amplitude which causes full saturation in case of an ideal clipper ($p \to \infty$). With this substitution, the nonlinearity depends on the ratio A/A_0'. Since the output is related to the saturation level, the equation is decoupled from absolute signal levels and the actual value for the small signal gain v. This allows performance comparisons for different drive levels independent from circuit-related absolute power levels and the gain v.

For the definition of the *input power backoff* (IBO), we can take $P_0' = (A_0')^2$ as the input reference level in this work. We define IBO as the ratio of the reference level to the average input power P_{in}. Together with (39), the definitions of IBO and OBO read

$$\begin{aligned} \text{IBO} &= P_0'/P_{\text{in}} = (A_0')^2/E\{A^2(t)\} \\ \text{OBO} &= P_0/P_{\text{out}} = A_0^2/E\{(g(A))^2\} \end{aligned} \tag{41}$$

Only for a constant input envelope, Equation (40) describes the relationship between IBO and OBO as shown in the graph in Figure 3 (with $A_{\text{sat}} \equiv A_0$ and $A_{\text{ref}} \equiv A_0'$). For amplitude-varying signals, the output power P_{out} depends on the probability density function (pdf) $P_A(A)$ of the radial component A.

$$P_{\text{out}} = E\{(g(A))^2\} = \int_{A=0}^{\infty} P_A(A)(g(A))^2 dA \tag{42}$$

In the following, the output power backoff for an OFDM signal is calculated from the input backoff. It is assumed that the inphase and quadrature components have equal power and are nearly Gaussian, so that the radial component can be approximated with a Rayleigh distribution given as follows, including mean and variance ([Joh01]):

$$\begin{aligned} P_A(A) = \frac{A}{\sigma^2}\exp(-\frac{A^2}{2\sigma^2}), &\quad \mu_A = E\{A^2\} = \sigma\sqrt{\pi/2} \\ \text{var}\{A\} = E\{(A-\mu_A)^2\} = \frac{4-\pi}{2}\sigma^2, &\qquad E\{A^2\} = 2\sigma^2 \end{aligned} \tag{43}$$

σ^2 is the power of each Cartesian component and $2\sigma^2$ equal to input signal power P_{in}. Combining (40)-(43) yields

$$P_{\text{out}} = A_0^2 \int_{A=0}^{\infty} \frac{A}{\sigma^2}\exp(-\frac{A^2}{2\sigma^2}) \left(\frac{A/A_0'}{(1+(\frac{A}{A_0'})^{2p})^{\frac{1}{2p}}} \right)^2 dA \tag{44}$$

$$\Leftrightarrow 1/\text{OBO} = P_{\text{out}}/A_0^2 = \int_{A=0}^{\infty} \frac{A}{\sigma^2}\exp(-\frac{A^2}{2\sigma^2}) \left(\frac{A/A_0'}{[1+(\frac{A}{A_0'})^{2p}]^{\frac{1}{2p}}} \right)^2 dA$$

$$= \frac{(A_0')^2}{\sigma^2} \int_{A=0}^{\infty} \frac{A}{A_0'}\exp(-\frac{A^2}{(A_0')^2}\frac{(A_0')^2}{2\sigma^2}) \left(\frac{A/A_0'}{[1+(\frac{A}{A_0'})^{2p}]^{\frac{1}{2p}}} \right)^2 \frac{dA}{A_0'}$$

$$\Leftrightarrow 1/\text{OBO} = 2\cdot\text{IBO}\cdot\int_{\alpha=0}^{\infty} \alpha\exp(-\alpha^2\cdot\text{IBO})\cdot\frac{\alpha^2}{[1+\alpha^{2p}]^{\frac{1}{p}}}d\alpha$$

For the ideal limiter with $p \to \infty$, calculation is straight-forward by splitting the integral into two parts and performing integration from $\alpha = 0$ to $\alpha = 1$ and $\alpha = 1$ to $\alpha = \infty$. This yields

$$\mathrm{OBO} = \frac{\mathrm{IBO}}{1 - e^{-\mathrm{IBO}}} \tag{45}$$

For typical values of p in the range of 1 to 4, an analytical solution for the integral (44) is not easily found. It has been calculated with numerical integration using adaptive Gauss-Kronrod quadrature method.

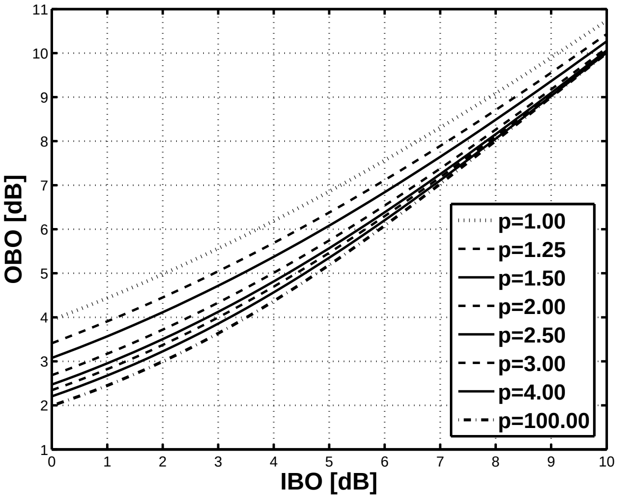


Figure 4: IBO versus OBO for Rapp amplifier model and Rayleigh-distributed input amplitude

The obtained graph is useful for performance evaluation in combination with a link budget analysis. In the data sheet of PA's one can often find two specifications, the output saturation power P_{sat} and the 1-dB-compression point P_{1dB}. The latter is defined as the input power level for which the output power level is one dB lower than the ideal value on the linear curve, due to saturation. With P_{sat}, P_{1dB} and the small signal gain v, it is possible to determine the smoothing parameter p and fit the Rapp model (40).

If the Rapp model does not adequately fit the amplifier curve, a more general approach is to use polynomial functions for AM/AM and AM/PM conversion. In order to fit the polynomials, thorough measurement is required. The Rapp model can also be modified in the way as to include any AM/PM conversion.

In this work, it is assumed that no significant distortion arises in the receiver. In this case, the transmitter PA is regarded as the main contributor to nonlinear distortions in the whole link. In general, nonlinear distortion leads to a widening of the signal spectrum, which is called *spectral regrowth* ([Beh98]). Radio standards impose restrictions in the co-channel interference through the specification of a spectral mask. The widened and perhaps filtered signal spectrum must not violate the spectral mask. Vice versa, in the design of a physical layer, the specification of the occupied bandwidth (the signal bandwidth covering 99% of the power) and spectral mask needs to take into account the consequences for the power amplifier design and operating point. The use of a subsequent filter introduces a loss of transmission power, which lowers the link budget.

2.6 I/Q mismatch

We first recall the classical problem of *frequency-independent I/Q mismatch* ([Beh98]). In real-world analog quadrature modulators and demodulators, the phase shift of the VCO signals for the inphase and quadrature path is not exactly 90 degree as required in equation (1). In addition, the gain in each path will differ in some amount. Let's first assume that the transmitter generates a perfect signal and that the receiver front end creates imbalance. In complex notation, perfect frequency shift is accomplished by multiplication with $x_{\text{vco,ideal}}(t) = \exp(-j2\pi f_c t)$. Phase and gain imbalance are modeled with a modified oscillator signal

$$x_{\text{vco}}(t) = \cos(2\pi f_c t) - jg\sin(2\pi f_c t + \phi) \tag{46}$$

where g and ϕ represent gain and phase imbalance. The ideal VCO signal would be obtained for $\phi = 0$ and $g = 1$. Upconversion of the I/Q transmit signal $z_{tx}(t)$ with an ideal modulator and downconversion with an imperfect demodulator with subsequent low-pass filtering results in the equivalent I/Q operation ([Mar05])

$$\begin{gathered} z_{rx}(t) = K_1 \cdot z_{tx}(t) + K_2 \cdot \overline{z_{tx}(t)} \\ K_1 = \tfrac{1+ge^{-j\phi}}{2} \quad K_2 = \tfrac{1-ge^{+j\phi}}{2} \end{gathered} \tag{47}$$

The complex conjugate term $\overline{z_{tx}(t)}$ represents the baseband signal flipped in frequency domain, so that all positive frequencies are mirrored to negative ones and vice versa. I/Q mismatch creates self noise of the received signal. The amount is given by the frequency crosstalk coefficient K_2.

We briefly discuss the impact of I/Q mismatch on OFDM reception[1]. In general, the problem is a serious one, since subcarriers received with high channel gain may strongly disturb weakly received subcarriers in case of frequency selective channels. Without any carrier frequency offset (CFO) between transmitter and receiver, the crosstalk would only result in mutual interference of subcarriers with opposite frequencies $\pm k\Delta f$. In other words, the mirror signal $\overline{z_{tx}(t)}$ fulfills the orthogonality condition, because all mirrored subcarriers are still on the same frequency grid $n\Delta f$, where Δf denotes the subcarrier spacing. Unfortunately, carrier frequency offset, which is almost always present in practice, destroys the orthogonality of the mirror signal for offsets not in the vicinity of $n\Delta f$.

Many attempts have been made to solve the imbalance problem in frequency domain ([M. 04], [Mar05], [J. 03b], [F. 07b], [J. 09]). A blind compensation technique is presented in [M. 04], which achieves high accuracy after a long convergence time. The scheme relies on the assumption of a mutual subcarrier crosstalk just described. Therefore, for nonzero CFO, the performance of the blind compensation scheme quickly degrades. The same author presented a similar scheme in [Mar05], which utilizes the OFDM preamble for faster conversion. But in order to achieve immunity against CFO, half of the preamble subcarrier symbols had to be replaced with zeros. Such a restriction might not be acceptable for a radio standard.

---

[1] An introduction to OFDM is given in Section 4.1 and Appendix C.

Aware of the problem, authors in [J. 03b] present a scheme, where carrier frequency offset is taken into account. The scheme performs preamble-based joint- compensation of the CFO and I/Q imbalance. It is shown by simulation that the CFO, which needs to be estimated first, can be obtained with sufficient precision using a standard correlation technique ([T. 97]) even in the presence of IQ-imbalance. This was shown only for a maximum imbalance of 10° phase offset and 10% I/Q amplitude mismatch. Two FFT operations must be performed on the preamble symbol, and the algorithm involves channel estimation.

Another method, which also treats CFO and I/Q imbalance together, was presented in [F. 07b], operating on the 802.11a preamble. The algorithm restricts the maximum carrier frequency offset to 1 instead of 2 subcarrier spacings.

So far, the mentioned references just treated the case of frequency independent I/Q imbalance, the estimation of two parameters. Tolerances in the analog baseband filters may lead to additional *frequency-dependent I/Q-mismatch*. In this case, the problem is more involved. On the other hand, it can be expected that the low-pass filters do not change their characteristics over time. Therefore, digital compensation filters can be employed in time domain, which are calibrated just once during manufacturing process.

It may seem to be an efficient solution to perform the compensation of frequency dependent I/Q mismatch in frequency domain before IFFT operation in the transmitter, and after FFT operation in the receiver respectively. This is suggested in [J. 09]. Transmitter and receiver compensate their mismatch independently. Note that a joint compensation performed by the receiver is not only much more difficult, but also inadequate, because radio standards require the transmitter to ensure a certain signal quality in terms of error vector magnitude (EVM). Unfortunately, authors of [J. 09] ignored the problem of the CFO.

Frequency-domain compensation seems indeed to be an efficient way for the transmitter. For the receiver, I/Q compensation must come before frequency offset correction, since compensations must be performed in reversed order with respect to the effects. But normal OFDM FFT operation requires CFO compensation to be done first to avoid loss of orthogonality. Therefore, an additional FFT and IFFT operation would be needed for I/Q compensation prior to CFO correction and OFDM FFT operation.

A calibrated compensation of the frequency dependent I/Q mismatch leaves the problem of phase (and gain) imbalance to be solved. As mentioned, the transmitter-generated imbalance needs to be tackled at the source in order to satisfy EVM requirements. In the worst case, the receiver may experience a slowly time-varying phase and gain mismatch created in the demodulator. Application of the aforementioned techniques is paid either with considerably increased complexity or some restriction with respect to system architecture or preamble signal waveform.

To decouple the problem from normal receiver operation, I/Q compensation can be done via blind source separation in time-domain ([Val01]). In this work, I/Q-imbalance compensation has not been addressed. It is assumed that a two-step time-domain scheme is employed, which can compensate frequency dependent, time-invariant mismatch with calibrated filters and demodulator mismatch with adaptive techniques in time-domain. In the ideal case, I/Q mismatch is then sufficiently compensated and therefore transparent to the receiver.

2.7 Summarized link model, parameter investigation

Figure 5 shows the proposed equivalent baseband link simulation model in its general form, but excluding I/Q mismatch. This model includes all transmitter and receiver impairments, which have been described in the previous sections.

After signal quantization, an arbitrary resampling block is used to realize clock jitter and clock frequency deviation. This resampling is well approximated with a N-ary oversampling followed by peacewise linear interpolation. N=16 is sufficient in terms for precision for most cases. Note that the resulting waveform has almost the same sampling rate $f_{T,tx}$. A subsequent fractional resampling block recalculates the signal waveform for a rate change factor of $f_{T,ch}/f_{T,tx} = I/R$. This is done with interpolation by factor I using I-1 polyphase filters followed by R-ary decimation. The aim is to match the sampling rate to the rate used for the channel impulse response, but also to extend the bandwidth in order to account for spectral regrowth caused by the power amplifier. Alternatively, the channel impulse response can be resampled to match the (extended) signal sampling rate. The next block performs phase modulation to model CFO and phase noise, followed by PA saturation. After PA, the signal is convolved with a deterministic time-variant or static impulse response, see Appendix B.2 and B.3. Channel models are treated in Section 3.

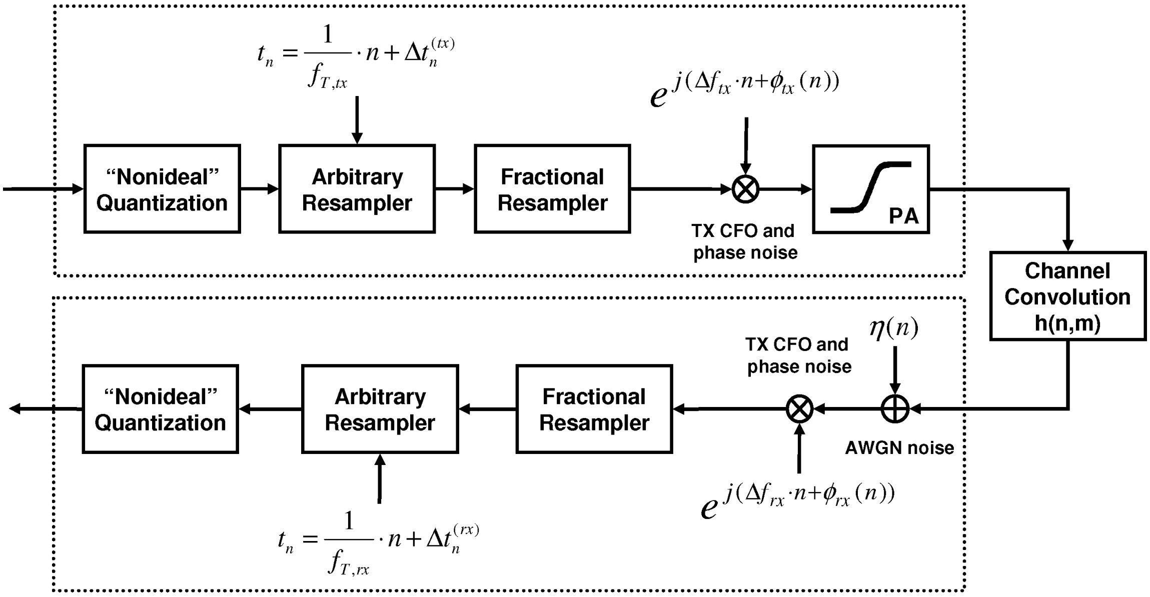


Figure 5: Equivalent baseband link model

The receiver first of all adds white noise to the signal, which reflects its limited sensitivity. The following blocks perform the same functions as for the transmitter, in reversed order. As already mentioned, one may combine clock deviation, jitter, phase noise and frequency offset from TX and RX into equivalent receiver effect blocks, leaving quantization, fractional resampling and the saturation block for the transmitter.

We shortly consider AWGN noise again. For simulation, the noise level P_N is given to $P_N = P_{\mathrm{sig}}/\mathrm{SNR}$. This "SNR" here is the ratio of the average signal level P_{sig} to the noise power in a bandwidth of $f_{T,ch}$, which is the underlying sampling rate when adding noise. A more meaningful measure is the inband-SNR, $\mathrm{SNR}_{\mathrm{sig}}$, the ratio of signal to noise in the bandwidth B_{sig}. $\mathrm{SNR}_{\mathrm{sig}} = \mathrm{SNR} \cdot (f_{T,ch}/B_{\mathrm{sig}}) \geq \mathrm{SNR}$, because noise outside the signal bandwidth can be filtered out. For a radio link, the inband SNR is determined by the signal bandwidth, received signal level and noise figure of the complete analog receiver front end. Inband SNR most often will be the appropriate measure used for synchronization performance evaluation. But for demodulation and decoding, the usual figure of merit is the performance of the system versus E_b/N_0, with bit energy E_b and noise spectral density N_0. The reason is that the bit error rate of coherent demodulation using matched filtering depends on this ratio. With $P_N = N_0 \cdot f_{T,ch}$ and $P_{\mathrm{sig}} = E_b \cdot R_b$, R_b being the (effective) information bit rate, the relation to SNR is

$$\begin{aligned} \mathrm{SNR} &= P_{\mathrm{sig}}/P_N = E_b \cdot R_b/(N_0 \cdot f_{T,ch}) \\ &= (E_b/N_0)(R_b/f_{T,ch}) \Leftrightarrow (E_b/N_0) = \mathrm{SNR} \cdot (f_{T,ch}/R_b) \end{aligned} \tag{48}$$

For very long frames, R_b is asymptotically equal to the PHY data rate, that is when neglecting preamble overhead. We can write (48) in a different way to attain a more general form. The energy of the sync pulse of two-sided bandwidth $f_{T,ch}$ is equal to T_{ch}, so that the total energy of the sampled noiseless frame $x(k)$ is $E_{\mathrm{frame}} = T_{ch} \sum_k |x(k)|^2$. For N_{info} information bits,

$$E_b/N_0 = \frac{T_{ch} \sum_k |x(k)|^2}{N_0 \cdot N_{\mathrm{info}}} = \frac{1}{P_N} \frac{\sum_k |x(k)|^2}{N_{\mathrm{info}}} \tag{49}$$

System parameters

For the narrowband OFDM system, the used D/A-converters had 14 bit resolution, whereas A/D converters were available with only 8 bit resolution ([FUJ04], [Nat05]). For this reason, DA converter quantization could be neglected. The typical ENOB for the A/D converter is given to 7.4 bit and seems not to depend on the input frequency. In the simulation the reduced ENOB has been modeled with an additional white noise source.

For the narrowband system, commercial VCOs operating at 800 MHz can be found with a phase noise of -105 dBc/Hz @ 10 kHz ([CRY]). We assume a value of -100 dBc/Hz @ 10 kHz for the TX/RX VCO in those simulations where the jitter effect is included.

Measurements of phase noise for fabricated SiGe oscillators in IHP technology have been published in [W. 04] and later improved ([F. 07a]). The measured values at given offset frequencies and the corresponding 3-dB double-sided linewidths are given in table 2. Whereas the first oscillator could be used for direct downconversion, the second one was intended for a sliding-IF superheterodyne architecture with an intermediate frequency of 12 GHz. Positions 3-4 just represent the case of two identical VCOs at transmitter and receiver, so that phase noise is increased by 3 dB and the linewidth is doubled. Throughout the subsequent system design considerations, we will assume a worst case value of $L_{ssb} = -87\ dBc/Hz$ at 1 MHz offset (pos. 3).

The parameters of the used IHP power amplifier has been reported in [C. 09]. Gain and phase distortion are shown in Figure 6. The smoothing parameter for the Rapp model in Equation (40) has been set to $p = 1.4$ to match the measured characteristics (Figure 7). The low phase distortion was neglected in the model.

Table 2: Measured phase noise values of IHP oscillators

position	carrier frequency	Single-sideband phase noise at 1 MHz offset	equivalent linewidth
1 ([W. 04])	60 GHz	-90 dBc/Hz	6,3 kHz
2 ([F. 07a])	48 GHz	-98 dBc/Hz	1 kHz
3	60 GHz	-87 dBc/Hz	12,5 kHz
4	48 GHz	-95 dBc/Hz	2 kHz

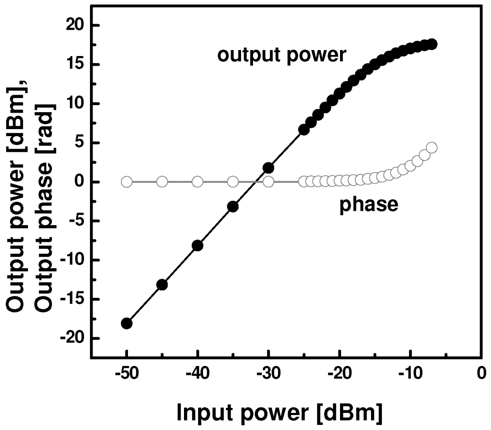


Figure 6: AM-AM and AM-PM characteristics of the utilized IHP SiGe BiCMOS power amplifier

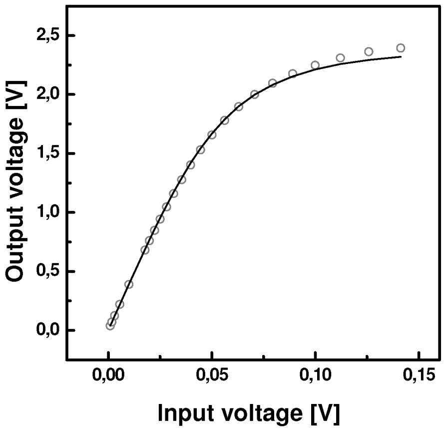


Figure 7: Comparison between measurement and the Rapp model (continuous line) with p = 1.4

3 Channel models for 60 GHz radio communication

The performance of radio communication systems like OFDM strongly depends on the typical characteristics of the faced radio channels. The system must cope with these characteristics in order to maximize performance in terms of goodput or quality of service. Therefore, a *channel model* is required for system design, which matches the typical experienced link scenarios as good as possible. A large set of different impulse responses is usually required for simulation in order to obtain meaningful results. In this way, the average system performance can be investigated, which hopefully covers most practical cases. Basically, there exist three common ways to obtain such a channel model viewed as a large set of impulse responses.

- One way consists in application of *ray tracing* applied on geometrical models of various environments. This method calculates the wave propagation for one or several different positions of transmitter and receiver. Frequency dependent reflexion coefficients of the reflecting and in some cases attenuating surfaces must be incorporated. In practice, construction of the environments and calculation of a large set of constellations can be very time consuming.
- Another method consists in conduction of a measurement campaign for a number of typical environments and for different transmitter and receiver locations. In such a campaign, advanced measurement techniques may be applied to determine the associated angles of the different propagation paths at the transmitter and/or receiver side.
- According to the third method, either ray-tracing simulations or direct channel measurements are performed in order to extract realistic fitting parameters for a *statistical* channel model.

Typically, a different channel response is needed for each iteration or number of iterations in system simulation. Since the outcome of the first two methods is a recorded set, each impulse response is just picked from this set. In contrast, a statistical model creates each response as a realization of a random process. Hence, the third method can deliver an arbitrary large set of channel responses.

Two channel models have been available for this work. The first model is the outcome of a measurement campaign performed by the Heinrich Hertz institute (HHI) in Berlin. This was done for IHP within the WIGWAM project. It will be referred to as the HHI model. The second model has been developed within the IEEE task group 802.15.3c for performance comparison between competing system proposals. It is a statistical model with distribution parameters obtained from measurements.

3.1 General characteristics of 60 GHz indoor channels

Before channel models are described in subsequent sections, we first investigate some published reports dealing with 60 GHz channel measurements in order to get an idea about the general propagation characteristics. The first thing to notice is that due to the short wavelength, free space loss at 60 GHz is considerably higher compared to 2.4 and 5 GHz systems. According to Friis equation, the received power P_r for transmitter power P_t, wavelength λ, transmitter and receiver antenna gains G_t and G_r and distance R is given to

$$P_r = G_t G_r \left(\frac{\lambda}{4\pi R} \right)^2 P_t \tag{50}$$

With $\lambda = c/f_c$, for $c \approx 3 \cdot 10^8 m/s$ denoting the speed of light and f_c being the carrier frequency, the loss term

$$L(R) = 20 \cdot \log_{10}[(4\pi R)/\lambda] \tag{51}$$

equals 68 dB attenuation at 1 meter and 88 dB at 10 meters. The high loss leads to wave propagation between transmitter and receiver, *which is dominated by a LOS path (if such is seen) and first and second order reflection paths* ([A. 09], [J.H97], [Hao02]). In comparison to 5 GHz, where richer scattering is observed, 60 GHz wave propagation has a quasi-optical nature, and the dominant propagation paths appear to be well predictable by ray-tracing ([A. 09]).

Friis equation applies only for ideal one path-propagation. For multipath-propagation environments, the path loss $L_{\mathrm{dB}}(d)$ (in decibel) for RX-TX distance d is often approximated as

$$L_{\mathrm{dB}}(d) = L_{\mathrm{dB}}(d_0) + 10n(d/d_0) \tag{52}$$

The first contribution arises from a fixed path loss for some reference distance d_0 from the transmitter. The other is then determined by the ratio d/d_0. If d_0 is chosen to one meter, Friis equation may still apply for $L_{\mathrm{dB}}(d_0)$. Parameter n is the average path loss exponent. A deviation from $n = 2$ indicates the influence of the other path components.

[Hao02] reports an extensive measurement campaign, which was carried out in and between various rooms, hallways and halls of a university, but also outdoors. The averaged path loss exponent for all locations together was calculated to $n = 1.88$. The rms delay spread (see Appendix B.5) for omnidirectional antennas [1] ranged from 7.5 ns to 76.5 ns for hallways and 10.9 ns to 41 ns for rooms, the largest room being about 60 square meters in size. In general, rms delay spread increases for larger room size as well as for larger TX-RX separation. Penetration losses of concrete walls are higher than 30-35 dB so that walls would act like cell boundaries of 60 GHz networks.

[J.H97] also performed measurements (with a signal bandwidth of just 200 MHz) with omnidirectional antennas, but only in one big room with a size of about 100 square meters. The transmitter had a fixed position and the receiver was taken to different locations within the room to obtain a variety of impulse responses. Authors neglected any path contributions outside a window of 100 ns. The paths with the highest delay were at least 35 dB less in power compared to the LOS path. NLOS measurements were also performed with a person taken as an obstacle for the LOS path. The average LOS attenuation was measured to 20 dB. We note that the human body is likely to be a frequent obstacle for home video applications with a 60 GHz radio link between a decoder and a television screen.

A recent publication [A. 09] gives some additional insight into 60 GHz wave propagation characteristics. Measurements with a signal bandwidth of 800 MHz were done in a rather small conference

---

[1] The omni-directionality in the horizontal plane was simulated with spin-measurements using high-gain antennas, see [Hao02]. In this way, additional angle-of-arrival measurements have been carried out.

room of size 3m x 4.5m x 3m (W x L x H). Antennas with high directivity were used for transmitter and receiver, which were steerable in both angles. In this way LOS, first and second order reflections could be well studied. It was found that first- and second order reflections are about 10-15 dB and 24-28 dB below the LOS path, respectively. First order reflections showed frequency flat behavior in some cases, but also frequency selective behavior in others. This is attributed to the *cluster phenomenon*, where each reflected ray actually consists of two to several ray contributions closely spaced in time. Frequency selectivity was found to be strongest for the ceiling reflection. The reason was that the typical ceiling was composed of a concrete and a suspended wall. Another important result was the high *cross-polarization discrimination (XPD)* measured at 60 GHz. This value describes an additional mean attenuation experienced at the receiver if its antenna has the opposite polarization with respect to the transmitter antenna. With a value of 20 dB it is considerably larger than at 5 GHz and may enable a simultaneous transmission of two independent radio signals using a horizontally *and* vertically polarized antenna at both sides.

3.2 HHI Channel model

Now we turn the focus on the two channel models used for this work. Within the WIGWAM project, the Heinrich Hertz Institute Berlin has conducted measurements in a conference room of size 8.70m x 5.70m x 3m with antenna separations of 2,5 to 6 meters. The measurements were done for LOS and NLOS conditions for two antenna configurations: in the first case, both the transmitter and receiver were equipped with an omnidirectional antenna, whereas in the second case, a directional Vivaldi antenna with a gain of G=12 dBi and a half-power beam width of 30 degree was used for the receiver. This type of antenna, depicted in Figure 8, radiates in horizontal direction. It has been selected for the WIGWAM demonstrator. The Vivaldi antenna was positioned to point towards the transmitter. A metal plate was used to block the LOS path for additional NLOS measurements. The transmitter had a fixed position, and measurements were obtained for 12 different receiver locations in case of the omni-omni-link and 14 positions in case of the omni-Vivaldi configuration. For each location, the receiver was moved along one of two perpendicular axes to produce a channel measure after every millimeter. In this way, 100 measurements were obtained for each axis. For the omni-Vivaldi link, only the axis towards the transmitter was traversed. An overview is given in table 3.

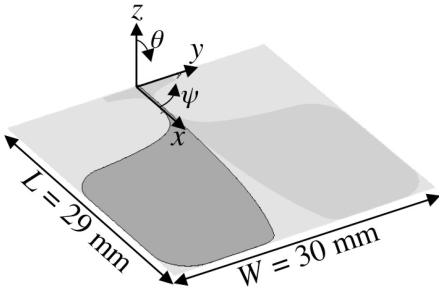


Figure 8: 60 GHz Vivaldi antenna used for the demonstrator

The channel responses at each of the 12 locations are correlated. Nevertheless, small scale fading

Table 3: Measured channel data

Set	transmitter antenna	receiver antenna	link condition	notes
1	omni	omni	LOS	2400 measured impulse responses at 12 room positions, 2 moving axes, 2x100 records per axis
2	omni	omni	NLOS	same as in 1
3	omni	Vivaldi	LOS	1400 measured impulse responses, 14 positions, 1 moving axis = 100 records
4	omni	Vivaldi	NLOS	same as in 3

causes variations of the frequency response when the receiver is moved, since many wavelengths are traversed. Channel data was delivered for 800 MHz bandwidth and cannot be used for the EASY-A wideband OFDM mode. Path arriving angles were not covered in this measurement campaign. Measured data was delivered in such a way that original amplitude relationships between all channels have been preserved. This may allow a link budget analysis for this particular room entirely based on measured data in contrast to common large-scale approximations. System performance can then be obtained for a given TX-RX distance.

An alternative is to separate small-scale from large-scale fading and handle large scale fading with a path loss law given in Equation (52). A very optimistic value of $n = 1.33$ for the path loss exponent was reported in [M. 09], based on similar measurements, but with a different antenna hight. For this work, it was decided to jointly normalize the received signal energy of the impulse responses, which belong to the same large-scale position, to unity. The average gain is then given by (52). For two reasons, this simplification is only a approximation. Firstly, it neglects that the Rician factor, the ratio of the LOS power to scattered power, will decrease with rising distance. Also, for very small distance of 2 meters, the position variations of ± 2 meters are already large scale and will produce fluctuations of the overall received power. Nonetheless, joint-normalization has been made for simplicity.

Table 4: RMS delay spread of HHI channel for 400 MHz bandwidth

omni-omni-LOS	omni-omni-NLOS	omni-Vivaldi-LOS	omni-Vivaldi-NLOS
5.0 ns	6.4 ns	3.1 ns	6.1 ns

For the four different scenarios, the mean rms delay spread has been calculated according to (244) in Appendix B.5 and is given in table 4. The cumulative distribution of the rms delay spread is plotted in Figure 9. This has been done after downsampling to 400 MHz bandwidth, the DFT bandwidth of the first OFDM demonstrator. Due to the small room size, the obtained delay spread values for this particular are quite small. The Vivaldi-antenna further reduces scattered components.

Note that the calculated delay spread gives only a rough characterization of the channel. Only if the shape of the power delay profile is known and depends on one parameter, the channel is fully characterized. This is the case for a WSSUS model with exponential power profile. In the current case,

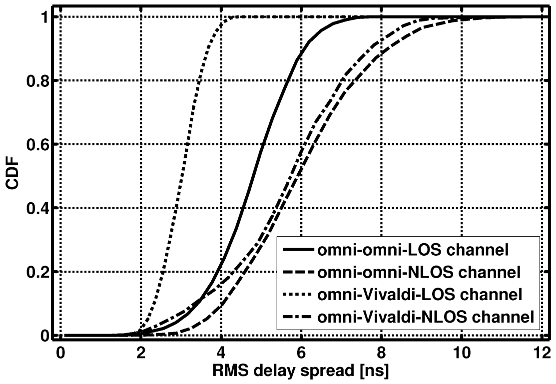


Figure 9: CDF of rms delay spread for HHI impulse responses downsampled to 400 MHz bandwidth

the effective inter-symbol-interference seen by an OFDM system can be obtained in a different way, as described in Section 4.3.1.

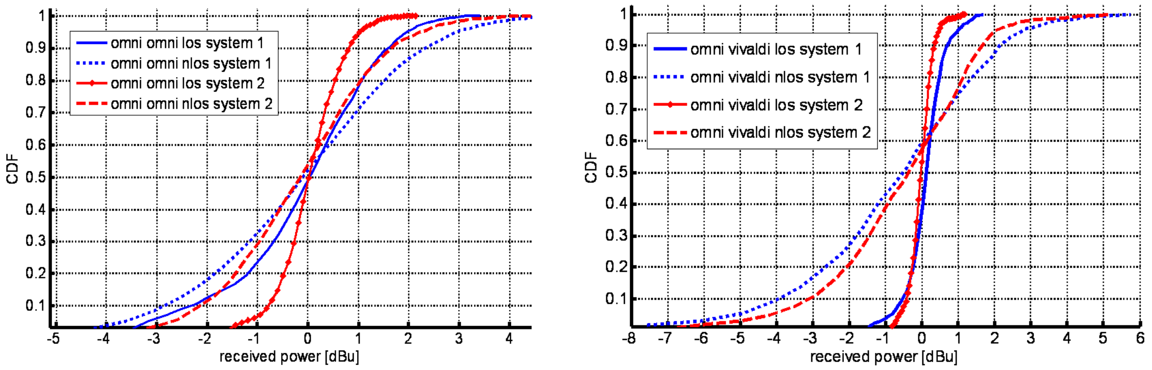


Figure 10: CDF of received signal level for 400 and 800 MHz bandwidth (HHI model)

Figure 10 plots the distribution of received signal power in 400 MHz ("system 1") and 800 MHz bandwidth ("system 2"). For the omni-omni-link, a system operating at 800 MHz has a clear advantage in suffering from less power fluctuations (left graph). The directivity of the Vivaldi antenna flattens the channel and effectively reduces such power fluctuations by attenuation of strong scatterers (right graph). The difference of the two systems is much less pronounced.

3.3 TG3c channel model

The IEEE task group 802.15.3c has focused on the standardization of a 60 GHz system for WPAN applications. For system comparison of different proposals, the interest group developed a channel model for various indoor scenarios, which is available as MATLAB code ([Hir07]).

Before introducing this model, it seems enlightening to provide a quick retrospective of the 5 GHz HIPERLAN model used in IEEE 802.11a ([ETS98]). Due to the fact that wave propagation at 5 GHz usually appears in the nature of rich scattering, the WSSUS channel model had been chosen for HIPERLAN. Five different channel scenarios A-E have been defined which are represented with specific power-delay profiles. These profiles are given in discrete-time domain for a sampling rate of 100 MHz. Each actual channel response $h(n)$ is generated as a realization of a random process. Assuming that a large number of superpositioned waves from different scatterers contributes to each tap $h(n)$, this tap is drawn from a complex Gaussian distribution with average power $P(n)$ defined by the corresponding power-delay profile. The taps $h(n)$ are assumed to be statistically independent. Note that a fixed time grid is merely an approximation, since waves will arrive with arbitrary delays. But this approximation is "good enough" for a signal bandwidth of 20 MHz, and statistical dependencies of adjacent taps are created when the channel response $h(n)$ is lowpass-filtered and downsampled to 20 MHz.

Due to the quasi-optical nature of indoor radio wave propagation at 60 GHz, the WSSUS model cannot be applied. Multipath propagation is dominated by the LOS path, if such exists, and reflections of low order, mainly first and second. Furthermore, for the considered signal bandwidths, the assumption of rich scattering in each channel tap does not hold. Therefore, a different model was needed. The taskgroup 802.15.3c has developed a model based on the one introduced by Saleh-Valenzuela in [Ade87]. This model seems appropriate, since it can generate impulses with a few dominating paths and it reproduces the cluster phenomenon at 60 GHz, i.e. reflections from obstacles are observed to arrive in clusters. The original Saleh-Valenzuela (SV) model has been modified to better fit the measurement statistics. The modifications mainly consist in introducing a Rician factor, and in a special treatment of the LOS path, which will be discussed later.

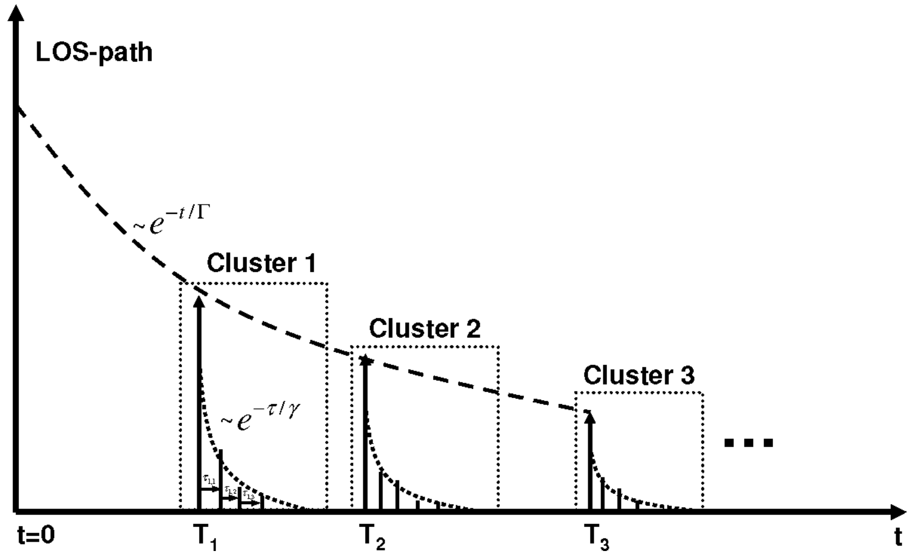


Figure 11: The modified Saleh-Valenzuela (SV) model applied to 60 GHz channels

Figure 11 illustrates the generation of a channel impulse response. This happens in an entirely different way compared to the WSSUS model. Clusters are supposed to arrive according to a *Poisson process* with a fixed average arrival rate of Λ. The conditional probability density function for the arrival time T_{l+1} of the first ray of the $(l+1)$-th cluster after the first ray of cluster l is given by an exponential distribution.

$$p(T_{l+1}|T_l) = \Lambda \cdot \exp(-\Lambda \cdot [T_{l+1} - T_l]) \tag{53}$$

In the same way, the rays of any cluster are also assumed to follow a Poisson process with arrival rate λ and arrival time probabilities

$$p(\tau_{l,q+1}|\tau_{l,q}) = \lambda \cdot \exp(-\lambda \cdot [\tau_{l,q+1} - \tau_{l,q}]) \tag{54}$$

The delays $\tau_{l,q}$ refer to the cluster arrival time T_l. So far, only the delay distribution law has been stated. Regarding the amplitudes, both clusters and rays within clusters are supposed to follow an exponentially decaying power law with decay parameters Γ and γ. The average cluster amplitude is proportional to $exp(-t/\Gamma)$, and the average amplitude of each ray (l, q) is proportional to $exp(-T_l/\Gamma)exp(-\tau_{l,q}/\gamma)$. The first path of each cluster is treated like a "local LOS path". An additional factor ΔK is used to balance the power between the expected cluster power level and the expected (mean) LOS power. Accordingly, a factor Δk is used to balance the power between the first cluster ray and all subsequent rays, like in a Rician distribution. Cluster and ray amplitudes are treated as being log-normal distributed. This requires the additional specification of the two standard deviations. The ray phases are assumed to be uniformly distributed in $[0, 2\pi)$. Angles of arrival are created via a Laplacian distribution, where the angle spread σ_ϕ is used as a model parameter. The obtained continuous-time channel responses, consisting of a large set of delayed Dirac-impulses representing the path components, can be weighted with some antenna pattern and finally resampled to match the sampling rate of the envisaged radio system. An detailed specification of the model is found in ([Hir07]). To account for desktop scenarios where the LOS path is accompanied by a strong desk reflection, the LOS path is subject to a two-path power distribution law, and the resulting model is named as the two-path Saleh-Valenzuela model (TSV).

Table 5: Selected TG3c channel models

Model	Scenario	TX antenna beam width	TX/RX distance	Λ [1/ns]	λ [1/ns]	Γ [ns]	γ [ns]
CM1.1 / CM2.1	Residential	360°	5 meters	0.191	1.22	4.46	6.25
CM1.2 / CM2.2	Residential	60°	5 meters	0.194	0.9	8.89	9.17
CM1.3 / CM2.3	Residential	30°	5 meters	0.144	1.17	21.5	4.35
CM3.1 / CM4.1	Office	30°	1-5 meters	0.041	0.971	49.8	45.2
CM3.2 / CM4.2	Office	60°	1-5 meters	0.027	0.293	38.8	64.9

A measurement campaign had been launched to perform a large number of measurements, which were primarily used to extract model parameters for different scenarios. The resulting channels are labeled as CMx.y, where the first digit represents the scenario and the second the selected transmit antenna beamwidth. CM1.x and CM2.x denote the LOS and NLOS models for the residential scenario, and CM3.x and CM4.x the LOS and NLOS models for the office scenario[1]. The most important

---

[1] A channel model CM3.3 has not been defined. The defined *library model* CM5 is not mandatory for system comparison and has been left out from consideration.

parameters of the channels CM1.x-CM4.x are given in table 5, excluding angle spread and other TSV parameters. The NLOS models CM2.x and CM4.x have the same SV parameters as CM1.x and CM3.x. The only difference consists in the missing LOS path (LOS extraction). The channels CM1.3, CM2.3 and CM3.1 have been selected as a mandatory subset among all available channels for system comparison. For these channels, the TX antenna has a 30° beamwidth.

The TG3c channel models are defined for different transmitter (TX) beamwidths, leaving the choice of the receiver (RX) antenna beam width open. To demonstrate the impact of the RX antenna, the distribution of the received input power and rms delay spread has been obtained for two cases: an omnidirectional and a high-gain antenna employed at the receiver. The results are based on a large set of model-generated impulse responses for a sampling rate of 2.16 GHz.

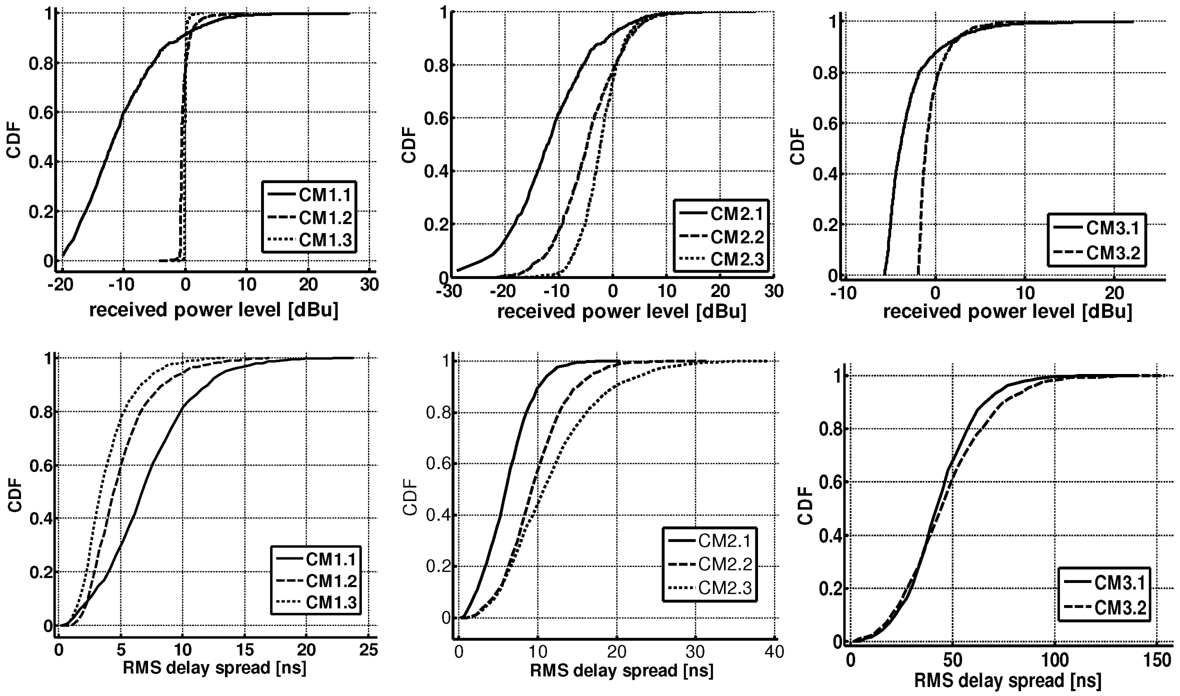


Figure 12: CDF and delay spread distributions for a receiver omni-antenna and 2.16 GHz bandwidth

Figure 12 presents these statistics for the omnidirectional case. For each channel model, joint normalization of received power has been done in such a way as to obtain an average received power level of unity (= 0 dBu), but preserving overall gain variations of individual impulses. CM1.1 employing an omnidirectional transmitter antenna shows extreme power fluctuations, which result in heavy performance degradation. CM3.1 and CM3.2 yield the highest delay spread.

According to the TG3c selection criteria document [IEE07], it was actually allowed to normalize each impulse response. But if doing so, the frequency diversity effect would not be taken into account, and a narrowband system seeing a flat channel without power fluctuations would always perform best. On the other hand, it is found that for CM2, a very few channel realizations have a very high power gain beyond 10-15 dB. These channels effectively lower the average power level for most of the channels leading to pessimistic performance results. To overcome this, we can indeed either normalize each response, so that performance is given with respect to receiver SNR, or discard outliers with a power gain exceeding a high threshold. The corresponding results for a receiver employed with a directive RX antenna are shown in Figure 13. The RX beam width has been assumed identical to that of the transmitter, see table 6. This would be the case when communication takes place between identical devices. For this reason, channels CM1.1 and CM2.1 are left out, because they represent the omni-directional case, which has been covered before. The rms delay spread distributions are plotted in the upper graph and the power distributions in the lower graphs, where CM2 is shown separately. This model is accompanied by a "lifted" version, where channels with more than 20 dB gain for CM2.2 and more than 10 dB gain for CM2.3 are left out.

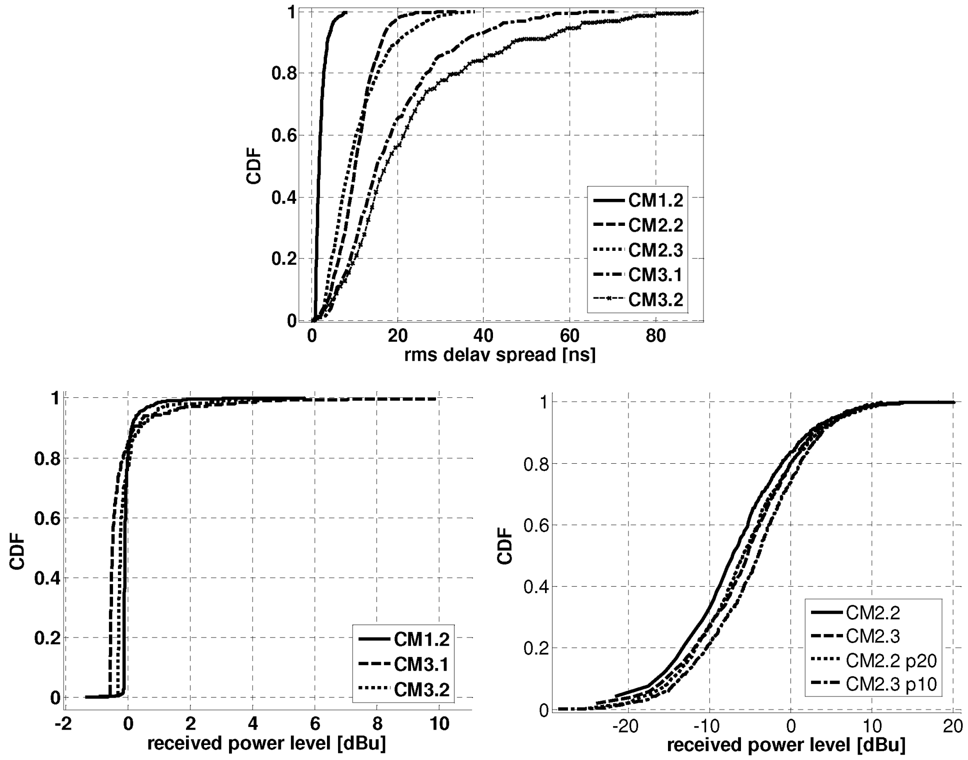


Figure 13: CDF and delay spread distributions for RX high-gain antenna and 2.16 GHz bandwidth

Table 6: Chosen RX antenna beam widths for selected TG3c channels

model	TX antenna beam width	RX antenna beam width
CM1.2	60°	60°
CM2.2	60°	60°
CM2.3	30°	30°
CM3.1	30°	30°
CM3.2	60°	60°

4 OFDM under non ideal link conditions

This chapter recalls some basics of coherent OFDM transmission using BPSK, QPSK and M-QAM mapping. In addition, it recapitulates the performance degradation of OFDM due to the channel delay spread, carrier frequency offset, phase noise and clock frequency offset. We quickly discuss the mitigation of phase noise in OFDM systems by application of a wideband PLL. Finally, we reconsider preamble based carrier frequency offset estimation using a generalized estimator and investigate the performance in presence of AWGN and Wiener phase noise. This chapter can be regarded as a preparation for the next, where system parameters and algorithms are discussed.

4.1 OFDM modulation

Conventional single carrier modulation schemes face the problem of inter-symbol interference (ISI) when the symbol time is reduced to the order of the channel delay spread. Looking at the problem in frequency domain, the transfer function of the radio channel appears *flat* for slow symbol rates associated with small signal bandwidths B_{sig} well below the coherence bandwidth B_{CH} and *frequency selective* for high symbol rates causing high signal bandwidths B_{sig} well exceeding B_{CH}. The conventional solution is to use an equalizer operating in time domain to compensate ISI, but the computational complexity of this equalizer significantly increases as the symbol time is further shortened in order to achieve higher rates.

OFDM is a powerful and elegant transmission technique used to cope with multipath radio propagation at high data rates where the radio channel appears frequency selective. In OFDM, the data stream is divided into many substreams, which are transmitted on narrowband frequency channels or *subcarriers*. The system parameters are chosen such that the symbol duration for these substreams is well above the maximum expected channel delay spread. In this way, complex equalization for each substream is avoided. In the following, only "traditional" OFDM with *guard interval* is considered, since it is the scheme which can be implemented with the lowest complexity. In Appendix C, an introduction to OFDM is given from a mathematical point of view. Some main points are summarized below.

- OFDM splits a high-rate data stream onto slow-rate substreams to establish a long symbol duration T_s, which should be large compared to the expected delay spread. This allows to extend the transmission pulses by a guard interval T_g to avoid ISI. The guard interval is short compared to the main pulse to cause only a slight performance degradation. Modulation, demodulation and finally channel equalization can be performed efficiently with the aid of FFT operation.

- The symbol time cannot be made arbitrary large, because a simple receiver assumes steady-state conditions during one OFDM symbol. Hence, the symbol time T_p should be well below the coherence time of the channel. Carrier phase and clock timing should be nearly constant during one symbol. OFDM is vulnerable against RF impairments, especially phase noise and residual carrier frequency offset, which lead to some loss of orthogonality. This vulnerability rises with higher symbol duration.

- According to the central limit theorem, the time-domain signal waveform approaches a complex Gaussian distribution for large numbers of subcarriers. OFDM, therefore, requires a power amplifier with high linearity at the operating point.
- Some subcarriers may experience strong fading caused by destructive interference of different paths. In result, the retrieved symbols on these subcarriers might be very noisy or even be below noise level, although the average channel SNR may be high. If simple hard-decision demapping of the subcarrier symbols without any form of channel coding would be used, a few badly received subcarriers may dominantly determine the bit error rate. To avoid this, redundancy must be introduced, and this is most efficiently done with the use of a channel code. Since reliability information of the retrieved symbols can be incorporated in the decoding decisions, the coding gain for a "properly designed" coded OFDM scheme is in general very large compared to uncoded OFDM in a multipath environment.

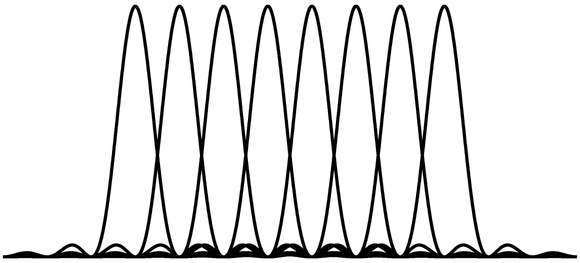


Figure 14: Overlapping power spectra of OFDM subcarriers

4.2 BPSK, QPSK and QAM-constellation mapping on subcarriers

A WLAN system usually aims to provide maximum data rate for a given link condition. Different data rates can be realized by varying the code rate or using different orders of modulation. A well established modulation type for OFDM is QAM-modulation where both phase and amplitude carry information. Although it does not represent the theoretical optimum and is paid with a slight *shaping loss*, it is widely used for many practical reasons. A mapping of 1,2,4 and 6 bits using BPSK, QPSK, 16-QAM and 64-QAM is depicted in Figure 94 in Appendix E. The inphase and quadrature component are treated independently. Gray-encoding (or gray-labeling) is used for the mapped bits in each dimension. This encoding scheme ensures that adjacent constellation points in inphase- or quadrature-direction differ only in a single bit. In this way, the uncoded bit error rate is minimized for high SNR, since symbol errors predominantly appear only between adjacent points. According to ([G. 98]), this is also true for bit-interleaved coded modulation (BICM). To establish an equal transmission power of unity for all modulations, symbols must be weighted with amplitude correction factors given in the table in Appendix E.

4.3 Degradation of OFDM due to imperfect synchronization and RF impairments

In comparison to single carrier transmission, there exist basically two sources of interference for OFDM: intersymbol interference (ISI) and intercarrier interference (ICI). Seen from any subcarrier, ICI means self noise created by other subcarriers in the same OFDM symbol. This happens if orthogonality is partly lost. Such loss is caused by residual carrier frequency offset, phase noise, clock frequency offset, IQ mismatch and amplifier nonlinearity. OFDM is known to be especially sensitive with respect to phase noise and CFO. ISI involves interference generated by adjacent OFDM symbols and is usually caused by channel delays, which fall outside the guard interval. Note that this ISI is in fact ICI arising from subcarriers in adjacent OFDM symbols.

Analysis of the degradation of the various effects is covered by a large number of publications. In general, precise analytical prediction of receiver performance in presence of all detrimental effects is very difficult. Therefore, Monte Carlo simulation is often used to investigate or confirm system performance under real conditions. Most publications, which deal with analytical methods, consider each degradation effect separately in order to limit the complexity. In addition, an AWGN channel is not always, but often assumed to derive mathematically tractable results. A counterexample is given in [Shi03], where the bit error rate analysis of differentially encoded PSK for a frequency selective and time selective Rayleigh fading channel in the presence of frequency offset leads to closed form expressions covering several pages.

A common approach is to consider the variance of the additional "noise" contributions at symbol level for each subcarrier as a figure of merit for performance degradation. For any subcarrier, the "signal to noise ratio" is expressed as

$$\mathrm{SNR} = \frac{E_s}{N_0 + \sum_k \sigma_k^2} \tag{55}$$

E_s is the average energy of the useful signal, N_0 stands for the AWGN noise and σ_k^2 denote additional error terms arising from various interference. If each variance term σ_k^2 is small compared to N_0, (55) can be approximated to

$$\mathrm{SNR} = \frac{E_s/N_0}{1 + \sum_k \frac{\sigma_k^2}{N_0}} \approx \frac{E_s/N_0}{\prod_k (1 + \frac{\sigma_k^2}{N_0})} = \frac{\mathrm{SNR}_{\mathrm{ideal}}}{\prod_k (1 + \frac{\sigma_k^2}{N_0})} \tag{56}$$

with $\mathrm{SNR}_{\mathrm{ideal}} = E_s/N_0$. The total SNR degradation D in dB is then approximately equal to the sum of individual degradations D_k.

$$D = 10 \cdot \log_{10} \left(\frac{\mathrm{SNR}_{\mathrm{ideal}}}{\mathrm{SNR}} \right) \approx \sum_k 10 \cdot \log_{10}(1 + \frac{\sigma_k^2}{N_0}) = \sum_k D_k \tag{57}$$

This simple approach has its limitations. If the number of subcarriers is large and each additional interference term is well approximated by a complex Gaussian distribution, D may be a good figure of merit. In case that interference on some subcarrier arises predominantly from a few, typically adjacent, subcarriers, the distribution may strongly deviate from being Gaussian and the BER analysis cannot rely on this criterion. Furthermore, coded transmission may show a different behavior in the presence of interference compared to uncoded transmission. If additional interference is about the same for all but a few subcarriers (typically the ones at the edge of the signal spectrum), (57) is approximately valid for the whole received signal. This is often the case for a AWGN channel.

In the following, we examine only channel-induced ISI and the effect of residual carrier frequency offset, phase noise, and clock frequency offset. As mentioned, we assume no I/Q offset in this work. Effects of amplifier distortion will be investigated through simulation in a later section.

4.3.1 Residual intersymbol interference due to insufficient guard time length

In an optimized OFDM system, the cyclic prefix will not cover the total delay spread of the channel, since this would require a very long guard time. Instead, the guard time will typically be set to a value, which ensures a signal-to-interference ratio (SIR) just high enough as not to affect the required SNR for the highest modulation. The self-interference power generated through the dispersive channel can be calculated from the channel response, OFDM system parameters and the chosen timing position.

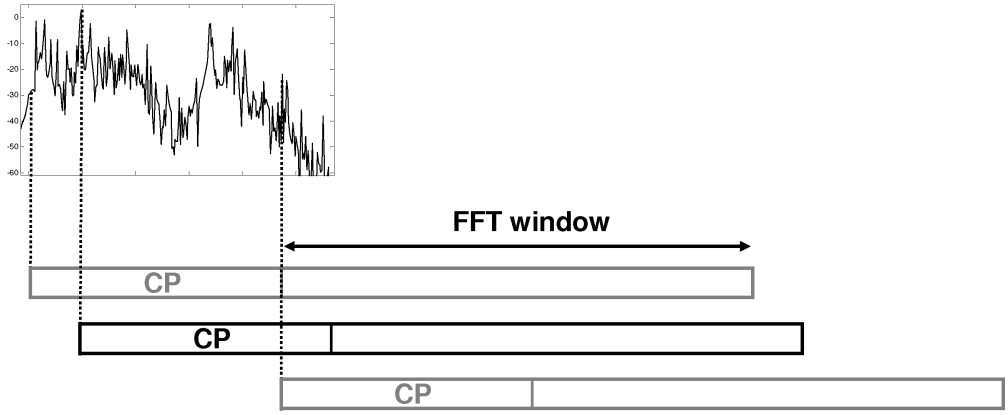


Figure 15: Channel dispersion inside and outside the guard interval

At first, we consider the general problem. For simplicity, a static channel and a rectangular pulse form are assumed[1]. The received signal arrives as a superposition of weighted and delayed transmit signal waveforms (Equation (229) in Appendix B). T_g denotes the guard time and T_s the DFT symbol time. The receiver may perform DFT operation for some OFDM data symbol starting at time T_w. All arriving signal copies of this symbol, which begin at $t_s \in [T_w - T_g, T_w]$ and finish at $t_e = t_s + T_s + T_g$, are correctly analyzed, because the DFT window lies inside the symbol waveform (see Appendix C). This is shown in Figure 15. The effect of a time offset for each copy is a nonzero phase slope seen in the frequency domain, but orthogonality is preserved. All other copies starting before or after the interval $[T_w - T_g, T_w]$ result in some ISI and ICI. A formula is needed to calculate the additional error variance. Such a formula is found in [M. 99], but the missing derivation is given in some unavailable reference. In the following, a simpler formula is derived, which proves to be sufficiently accurate in practice. The assumption is that the channel is at least short enough to affect adjacent OFDM symbols only.

---

[1] For slow varying channels, the considerations also apply.

The analysis is done in discrete-time domain. For sample duration T, the integer number of DFT samples and prefix samples is given to $N_s = T_s/T$ and $N_g = T_g/T$, respectively. The DFT in the receiver is taken for some intermediate OFDM symbol k at sample position $n = N_{w,k} = T_{w,k}/T$. We first think of a discrete-time Dirac channel $h_0(m) = \delta(m)$, which would be the case for the transmitter. In this case, the received cyclic prefix (CP) of this OFDM symbol would start at some time index $n = N_{0,k}$. It is useful to take this index as a time reference and define the *relative DFT start index* as

$$\Delta n_k = N_{w,k} - N_{0,k} - N_g \tag{58}$$

Note that Δn_k is constant if the frame timing (relative DFT position) is chosen constant for all symbols k, $N_{w,k} - N_{0,k} = \text{const}$, so that $\Delta n_k \equiv \Delta n$. For the beginning, we think of a static propagation channel seen at the receiver, and synchronized clocks. The impulse response composed by the channel and front end filters, shall be given in non-causal form as $h(m)$, $-\infty \leq m \leq \infty$. With constant frame timing for all symbols, there is a fixed set of $N_g + 1$ channel taps $h(m)$, which is correctly analyzed. According to (58), these are the channel taps with indices $m \in M(\Delta n) := [\Delta n, \Delta n + N_g]$, where $M(\Delta n)$ denotes the perfect analysis interval with respect to the channel response. Now we consider the channel taps outside this interval, which lead to interference. As will be shown, this interference depends on the *excess delay*, which can be defined as follows.

$$N_E(\Delta n, m) = \max\left\{\max(0, m - N_g - \Delta n), \max(0, \Delta n - m)\right\} \tag{59}$$

The reader can verify that $N_E(\Delta n, m) = 0$ for $m \in M(\Delta n)$, $N_E(\Delta n, m) = \Delta n - m$ for $m < \Delta n$ and $N_E(\Delta n, m) = m - N_g - \Delta n$ for $m > \Delta n + N_g$. The (positive) excess delay $N_E(\Delta n, m)$ for the waveform copy with amplitude factor $|h(m)|$ and delay m is nonzero if the copy arrives $N_E(\Delta n, m)$ samples too early or too late. It stands for the number of samples, which *leak* into the previous or next adjacent OFDM symbol, assuming that this symbol is processed with the same frame timing.

For the moment, we consider only one subcarrier with some index l. The leaking waveform causes interference, which is non-uniformly spread across the same and other subcarriers in the adjacent symbol. The equal number of samples is *missing* in the current symbol to complete the considered waveform copy for subcarrier l.

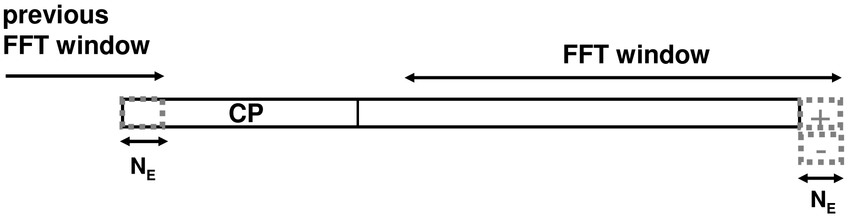


Figure 16: Sample leakage

Therefore, we can think of adding the missing samples to complete the copy and also subtracting them in order to keep the waveform the same, as shown in Figure 16. The subtracted waveform constitutes an error signal with the same energy as for the adjacent symbol. The only difference is the opposite sign and the correlation with the waveform of subcarrier l. Nevertheless, the created interference for the adjacent and the current symbol is approximately the same.

Until now we considered only one subcarrier. If we include all subcarriers, some will produce interference which partly leaks into the guard band outside the used signal spectrum, but this area is a small fraction of 10%-20% of the DFT spectrum. We also extend our view to all OFDM symbols (except the first and last symbol seeing less interference). P_{tx} shall denote the average signal power for a flat channel with unity gain. If we neglect that not the full bandwidth is used, the signal power is approximately equal to

$$P_{\text{rx}} \approx P_{\text{tx}} \cdot \left[\sum_{m=-\infty}^{\infty} |h(m)|^2 \right] \tag{60}$$

Every tap $h(m)$ with nonzero excess delay $N_E(\Delta n, m)$ produces interference energy of $P_{\text{tx}} \cdot |h(m)|^2 \cdot N_E(\Delta n, m)$ disturbing either the previous $(m < \Delta n)$ or the next symbol $(m > \Delta n + N_g)$. But since each data symbol suffers from ISI created from both neighbors, the total inter-symbol interference energy is given by

$$E_{\text{ISI}} = P_{\text{tx}} \cdot \left[\sum_{m=-\infty}^{\infty} |h(m)|^2 \cdot N_E(\Delta n, m) \right] \tag{61}$$

As stated, the inter-carrier interference within the same symbol is approximately equal to the ISI,

$$E_{\text{ICI}} \approx E_{\text{ISI}} \tag{62}$$

so that the total interference energy E_I is twice as large.

$$E_I = 2P_{\text{tx}} \cdot \left[\sum_{m=-\infty}^{\infty} |h(m)|^2 \cdot N_E(\Delta n, m) \right] \tag{63}$$

Hence, the signal-to-interference ratio (SIR) is equal to the ratio of perturbation-free signal energy $E_{\text{rx}} = P_{\text{tx}} N_s$ to interference energy E_I in time-domain. According to Parseval theorem, this ratio would be preserved for the subcarrier symbols in frequency domain if the full bandwidth was used.

$$\text{SIR} \approx E_{\text{rx}}/E_I = \frac{N_s \cdot P_{\text{tx}} \cdot \left[\sum_{m=-\infty}^{\infty} |h(m)|^2\right]}{2P_{\text{tx}} \cdot \left[\sum_{m=-\infty}^{\infty} |h(m)|^2 \cdot N_E(\Delta n, m)\right]} = \frac{\sum_{m=-\infty}^{\infty} |h(m)|^2}{2\left[\sum_{m=-\infty}^{\infty} |h(m)|^2 \cdot N_E(\Delta n, m)/N_s\right]} \tag{64}$$

Hence, the average power level of the additional interference is given to

$$\sigma_I^2 = 2P_{\text{tx}} \cdot \left[\sum_{m=-\infty}^{\infty} |h(m)|^2 \cdot N_E(\Delta n, m)/N_s \right] \tag{65}$$

Interestingly, Equation (65) states that interference can be reduced not only with a longer guard duration, but also by rising the DFT size N_s for a fixed guard duration.

The achieved accuracy in SIR estimation is shown in Figure 17 for 100 channel impulse responses for channel model CM3.2 and EASY-A OFDM parameters (Appendix G). This channel model has very high channel delay spread. For each response $h(m)$, the DFT window positions are chosen as to maximize the power profile area within the guard interval of $N_g + 1$ samples.

$$\Delta n = \arg\max_k \left\{ \sum_{m=k}^{k+N_g} |h(m)|^2 \right\} \tag{66}$$

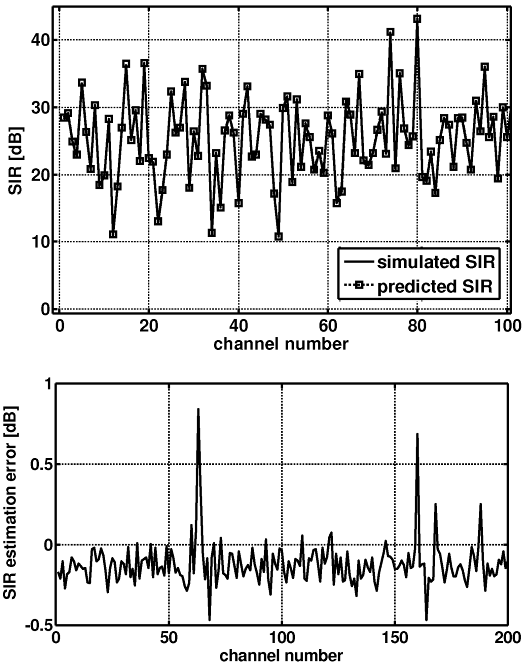


Figure 17: Simulated versus predicted channel SIR for CM3.2 and 118 ns guard time

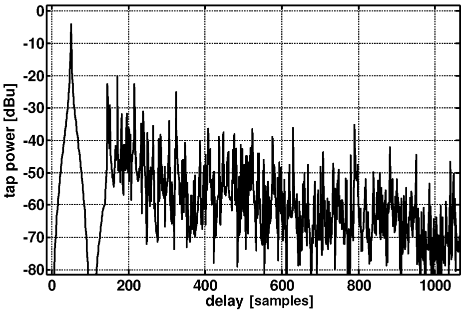


Figure 18: Power delay profile of a particular CM3.2 channel realization $h(m)$

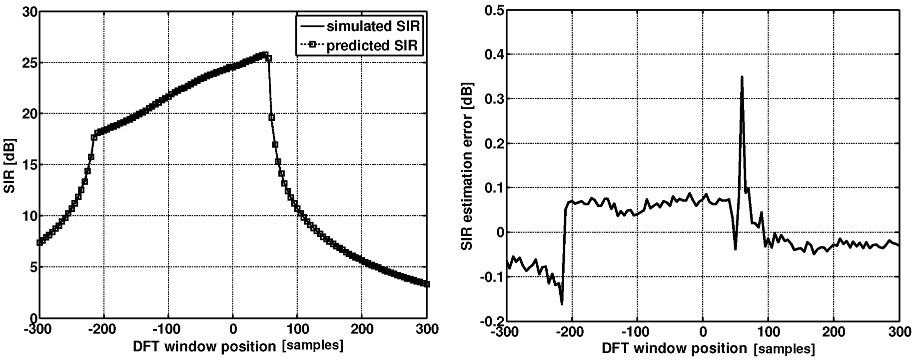


Figure 19: SIR prediction performance for varying frame timing Δn for channel $h(m)$

In Figure 18, the power delay profile (PDP) of some CM3.2 channel realization $h(m)$ is presented. The next two figures 19 demonstrate the SIR prediction performance in relation to the chosen window position for this particular channel. The guard interval consists of 256 samples. The SIR prediction error as the deviation between simulated and calculated SIR as a function of window position is given in the right figure. Again, similar results are obtained, since the estimation error stays below 0.4 dB.

It should be mentioned that this coincidence strictly holds only for the rectangular pulse. If windowing is used for OFDM, some prediction errors arise. Another point is the shape of the interference in frequency domain. The calculated SIR does not have to appear as flat noise in frequency domain. Therefore, the incorporated channel reliability of the bit metrics is negatively affected, because it is based on the assumption of white background noise, so that some additional performance degradation can be expected.

So far we have assumed a static channel $h(m)$. In the more general case, the channel seen at the receiver will be time variant. This happens due to a changing propagation channel or clock frequency deviation. In this case, equations (65) and (66) become time-variant. The receiver may have to adapt it's timing Δn_k for OFDM symbol k to attain best performance.

4.3.2 Degradation due to frequency offset and phase noise

Beside the high linearity requirements, CFO and phase noise sensitivity can be regarded as the major drawback of OFDM. This is especially the case when considering all-integrated analog transceivers for very high carrier frequencies, which cannot deliver very low phase noise values. Therefore, we need a quantification of the degradation in order to later decide on system parameters.

For analysis, one OFDM symbol consisting of N contiguous active subcarriers in an AWGN channel is considered (no zero subcarriers at DC). The received signal $x_{rx}(t)$ is a noisy and phase modulated version of the transmitted signal $x_{tx}(t)$.

$$x_{rx}(t) = x_{tx}(t)e^{j\phi(t)} + \eta(t) = \left[\frac{1}{\sqrt{N}} \sum_{n=-N/2}^{N/2-1} S_n e^{j2\pi n\Delta f t}\right] e^{j\phi(t)} + \eta(t) \tag{67}$$

The retrieved subcarrier symbol $\widetilde{S}_k$ obtained with correlation (DFT) reads ($T = 1/\Delta f$)

$$\begin{aligned} S_k &= \frac{\sqrt{N}}{T} \int_{t=0}^{T} x_{rx}(t) e^{-j2\pi k\Delta f t} dt \\ &= S_k \cdot \frac{1}{T} \int_{t=0}^{T} e^{j\phi(t)} dt + \sum_{n,n\neq k} S_n \cdot \frac{1}{T} \int_{t=0}^{T} e^{j\phi(t)} e^{-j2\pi(k-n)\Delta f t} dt + \frac{\sqrt{N}}{T} \int_{t=0}^{T} \eta(t) dt \\ &= S_k I_0 + \sum_{n,n\neq k} S_n \cdot I_{k-n} + v_\eta \end{aligned} \tag{68}$$

$$I_m = \textstyle\int_{t=0}^{T} e^{j\phi(t)} e^{-j2\pi m\Delta f t} dt \quad v_\eta = \frac{\sqrt{N}}{T} \int_{t=0}^{T} \eta(t) dt \tag{69}$$

I_0 is a common factor to all retrieved subcarrier symbols. The main effect is a phase rotation of the symbols by the *common phase error (CPE)*. An amplitude error also arises, but it is usually very small. A common method is to use pilot subcarriers for CPE correction. The middle term in Equation (68), composed of a weighted sum of all other subcarriers, represents inter-carrier interference (ICI), which appears as additional noise in each subcarrier. The last term is the contribution from additive noise power. Using the autocorrelation function of white noise, $R_{\eta\eta}(\tau) = N_0\delta(\tau)$, the subcarrier SNR without ICI is equal to $\mathrm{SNR}_{sc} = E\left\{S_k^2\right\}/E\left\{v_\eta^2\right\} = E\left\{S_k^2\right\}/(N_0\sqrt{N}/T) = (E\left\{S_k^2\right\}T/\sqrt{N})/N_0 = E_s/N_0$, with symbol energy $E_s = E\left\{S_k^2\right\}(T/\sqrt{N})$. Pollet has calculated the SNR degradation both for constant carrier frequency offset and Wiener phase noise ([T. 95]). The calculation was done assuming perfect CPE correction. The effect that outer subcarriers are subject to less interference is neglected. The formulas for SNR degradation given below are Taylor expansions and therefore valid for small degradations (up to 1 dB). The degradation in dB due to a frequency offset ΔF is given by

$$D_{dB} \approx \frac{10}{3 \cdot \ln 10} \cdot \left(\pi \frac{\Delta F}{\Delta f}\right)^2 \cdot \frac{E_s}{N_0} \tag{70}$$

The degradation depends on the ratio $\gamma = \Delta F/\Delta f$, the carrier offset normalized to the subcarrier spacing. For a more precise calculation at higher degradations, we rather use the additional noise variance σ_γ^2 given by

$$\sigma_\gamma^2 = \frac{1}{3}\left(\pi \frac{\Delta F}{\Delta f}\right)^2 \cdot E_s \tag{71}$$

The predicted degradation was verified by simulation for various values of γ, as shown in Figure 20 on the left side. In the diagram, $df \equiv \Delta F/\Delta f$. Following the wideband system parameters specified later, 828 subcarriers for a DFT size of 2048 have been used. With the large number of unused guard subcarriers, aliasing of ICI is avoided. On the right side, the BER degradation is given for uncoded 16-QAM modulation. As long as the SNR degradation D_{dB} stays below about 1 dB, the required SNR increase, which is needed to obtain the same BER performance as in the non-degraded case, is indeed roughly equal to D_{dB}.

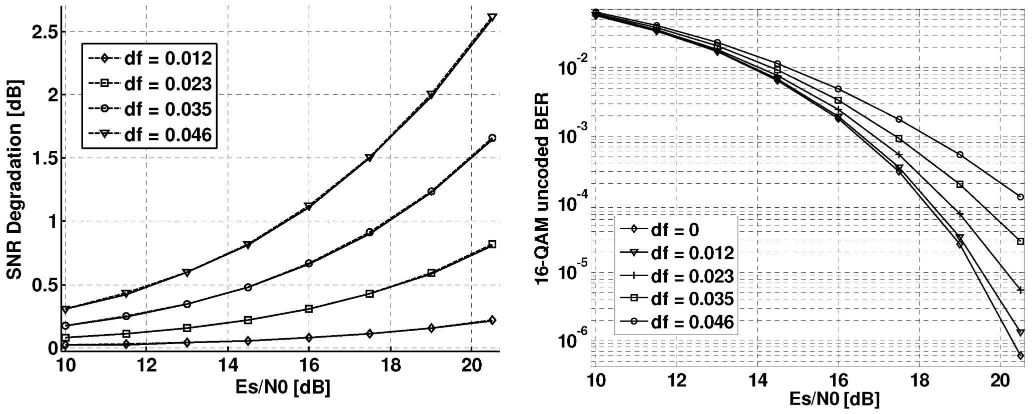


Figure 20: BER versus Symbol SNR degradation for uncoded 16-QAM due to carrier frequency offset

For small values, the SNR degradation D_{dB} due to Wiener phase noise with two-sided linewidth

Δf_{3dB} has been found to be ([T. 95]).

$$D_{dB} \approx \frac{11}{6 \cdot \ln 10} \cdot \left(\frac{2\pi \Delta f_{\mathrm{3dB}}}{\Delta f} \right) \cdot \frac{E_s}{N_0} \tag{72}$$

This degradation depends on the ratio $\beta = \Delta f_{\mathrm{3dB}} / \Delta f$, the two-sided VCO linewidth normalized to the subcarrier spacing. For a more precise calculation at higher degradations, we may rather make use of the additional noise variance σ_β^2 given by

$$\sigma_\beta^2 = \frac{11}{60} \cdot \frac{2\pi \Delta f_{\mathrm{3dB}}}{\Delta f} \cdot E_s \tag{73}$$

The degradation $D = 10 \cdot \log_{10}(1 + \sigma_\beta^2/N_0)$ was also verified for phase noise. Calculated versus simulated curves are shown in Figure 21 with $B \equiv \beta$. The calculated curves are only slightly above the simulated ones if degradation is small. Simulation was done for 828 subcarriers with a spacing of 2.11 MHz. The phase noise values correspond to single-sideband phase noise of L_{ssb} = -95 dBc/Hz, -92 dBc/Hz, -87 dBc/Hz and -85 dBc/Hz, respectively, measured at 1 MHz offset. Again, as long as SNR degradation is below 1 dB, the BER degradation is approximately equal. The exact BER dependence has been examined in [Luc98].

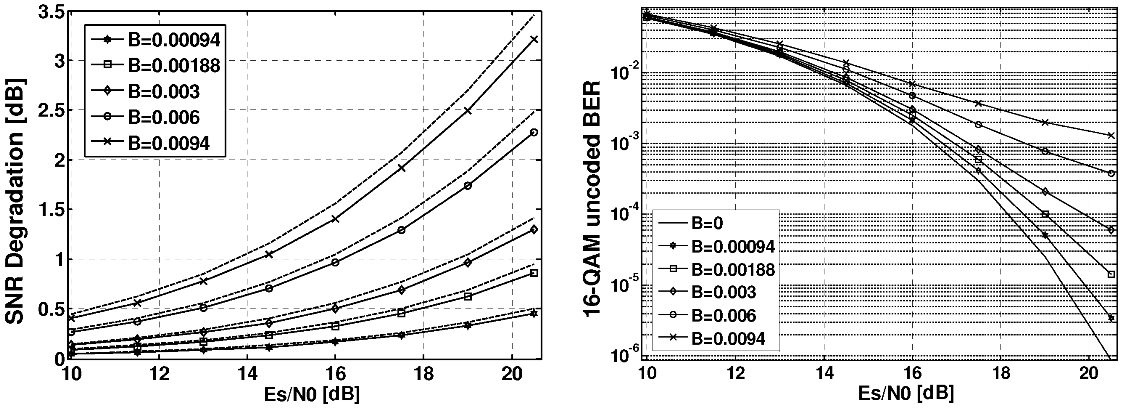


Figure 21: BER versus symbol SNR degradation for uncoded 16-QAM due to Wiener phase noise

In order to calculate the degradation for arbitrary phase noise spectra, the Stott model can be applied ([J. 98]). In this model, the phase change during the integration interval T is assumed to be small, so that the exponential phase noise factor can be approximated as

$$e^{j\phi(t)} \approx e^{j\phi(t=0)}(1 + j[\phi(t) - \phi(t=0)]) \tag{74}$$

Then each ICI-term for subcarrier k created by subcarrier n in Equation (68) can be written as

$$\begin{aligned} S_n \cdot \frac{1}{T} \int_{t=0}^{T} e^{j\phi(t)} e^{-j2\pi(k-n)\Delta f t} dt &\approx S_n e^{j\phi(t=0)} \cdot \frac{1}{T} \int_{t=0}^{T} (1 + j[\phi(t) - \phi(t=0)]) e^{-j2\pi(k-n)\Delta f t} dt \\ &= S_n \cdot e^{j\phi(t=0)} \cdot \frac{1}{T} \int_{t=0}^{T} j\phi(t) e^{-j2\pi(k-n)\Delta f t} dt \end{aligned} \tag{75}$$

Examination of (75) reveals that the phase spectrum is shifted in frequency domain by $-(k-n)\Delta f$ and then integrated over T. The same result is obtained if the frequency shifted phase passes a filter with impulse response $h(\tau) = (1/T)\Pi((\tau - T/2)/T)$ and is sampled at $t = T$[1]. The Fourier transform of $h(\tau)$ is equal to $\mathrm{Sinc}(fT)$. If the filter output is stationary, the variance of the output could also be obtained by integration of the resulting power spectrum $\Phi(f + (k-n)\Delta f)\mathrm{Sinc}^2(fT)$, but the same result is also obtained by integration of the shifted spectrum $\Phi(f)\mathrm{Sinc}^2([f-(k-n)\Delta f]T)$. The second term appears as a weighting function which mostly extracts spectral components around $(k-n)\Delta f$. This calculation could be repeated for the ICI contribution of each subcarrier. To simplify the procedure, a flat channel is assumed with equal power for each subcarrier. Apparently, the resulting power spectrum is then obtained by integration of the phase noise spectrum $\Phi(f)$ multiplied with a weighting function consisting of a large sum of shifted Sinc functions. This can be simplified if the ICI is over bounded by extending the number of subcarriers to infinity ([Denl2]). Using the property

$$\sum_{k=-\infty}^{\infty} \mathrm{Sinc}^2([f-(k-n)\Delta f]T) = 1 \tag{76}$$

the ICI power for every subcarrier n is finally given to

$$\sigma_{\mathrm{ICI}}^2 = E_s \cdot \int_{f=-\infty}^{\infty} \Phi(f)[1-\mathrm{Sinc}^2(fT)]df \tag{77}$$

The weighting function $W(f) = [1-\mathrm{Sinc}^2(f)]$ is the effect of the CPE correction, which acts as a high-pass filter and leads to finite noise contribution. Since phase noise is usually symmetrical around the carrier and monotonically decreasing with rising carrier offset, the middle carrier will suffer the highest phase noise. Hence, a tighter bound for the phase noise is given by the ICI of the middle subcarrier, obtained by

$$\sigma_{\mathrm{ICI}}^2 = E_s \cdot \int_{f=-B_{\mathrm{sig}}/2}^{B_{\mathrm{sig}}/2} \Phi(f)[1-\mathrm{Sinc}^2(fT)]df \tag{78}$$

In (78), integration is limited to the signal bandwidth B_{sig}. A constant noise floor can also be included, whereas (77) would wrongly give infinite ICI.

F. Herzel and the author have followed the idea to apply a wideband phase-locked loop (PLL) in order to reduce the experienced high phase noise of integrated VCOs ([F. 05]). A schematic view of the PLL is shown in Figure 22. The principle of this feedback circuit is to divide a VCO frequency f_{vco} by N and phase-lock the resulting oscillation with frequency f_{vco}/N to the reference oscillator. f_{vco} is equal to $f_{\mathrm{vco}} = N \cdot f_{\mathrm{ref}}$. Locking is accomplished with a phase discriminator and loop filter.

This case study considered a 56 GHz VCO, which was locked to a 54.6875 MHz reference oscillator. This reference frequency was chosen to obtain a division ratio of $f_{\mathrm{vco}}/f_{\mathrm{ref}}$ = 56 GHz/54.6875 MHz = 1024, which is easy to implement with frequency dividers. The effect of the PLL is to shape the phase noise spectrum of reference and VCO. More precisely, the phase noise of the reference is low-pass filtered whereas the VCO noise is high-pass filtered. In addition, the reference phase noise spectrum

---

[1] The rectangular function was defined in (28).

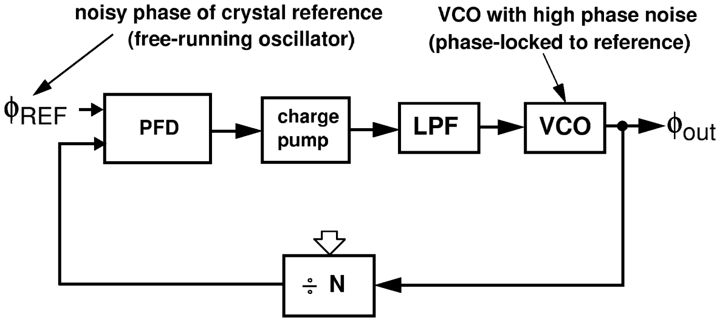


Figure 22: Schematic view of a integer-N charge-pump PLL ([F. 05])

appears amplified by N^2 at the higher frequency f_{vco}. Hence, the purity of the reference oscillator must be orders of magnitude better than the RF VCO, which is usually the case. If this VCO is used for frequency translation, the ICI after CPE correction in the baseband is given by ([F. 05])

$$\sigma_{\mathrm{ICI}}^2 = E_s \cdot \int_{f=-B_{\mathrm{sig}}/2}^{B_{\mathrm{sig}}/2} \left(\Phi_{\mathrm{vco}}(f) \cdot |1 - H_{\mathrm{LP}}(f)|^2 + N^2 \Phi_{\mathrm{ref}}(f) \cdot |H_{\mathrm{LP}}(f)|^2\right) [1 - \mathrm{Sinc}^2(fT)] df \quad (79)$$

The loop filter of an overdamped charge-pump PLL can be approximated with a first-order low-pass filter $H_{\mathrm{LP}}(j\omega) \approx \omega_l/(j\omega + \omega_l)$. The resulting ICI depending on PLL loop bandwidth is shown in Figure 23 for $L_{\mathrm{ssb,vco}}$ = -90 dBc/Hz at 1 MHz and different reference phase noise values ($E_s = 1$ and $\sigma_\phi \equiv \sqrt{\sigma_{\mathrm{ICI}}^2}$). Two graphs are shown to distinguish between simulation and analytical results. The second-order model considers the real transfer function of the PLL.

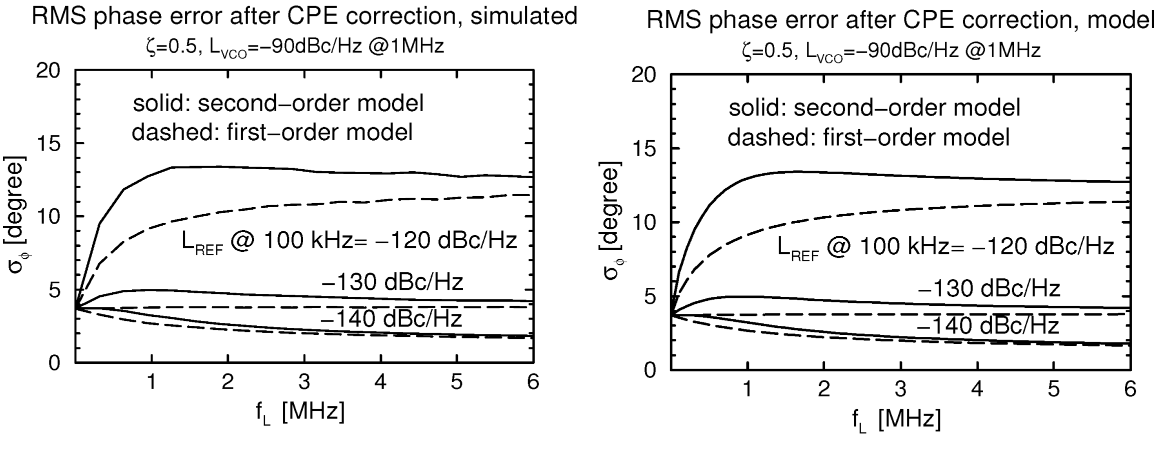


Figure 23: PLL output phase noise ([F. 05])

Due to the high frequency division-ratio of N=1024, the reference must have a high purity (-140 dBc/Hz at 100 kHz). According to this graph, the total phase noise could be reduced by a factor of two (6 dB) for loop bandwidths beyond 3 MHz. However, this has not been achieved in practice. Other various noise sources arising within the PLL circuit, which had been neglected in the model, lead to degraded noise performance. For this reason, a narrowband PLL with a noise spectrum similar to the free running oscillator, has been assumed for the VCO for later system simulations.

4.3.3 Degradation due to sampling clock frequency mismatch

A complex baseband implementation will usually work with a steady clock frequency. Very often no alignment to the transmitter clock is performed, because implementation of clock alignment is technically difficult. Since there will be some clock deviation, the timing drifts slowly out of synchronization during the frame reception. Two methods can be applied to counteract this. A standard solution for single-carrier modulation is to continuously resample the input stream ([Hei98]). The second method accepts some slight performance degradation in favour of a symbol-wise timing offset correction after FFT operation. This scheme is much easier to implement, because timing correction is equivalent to subcarrier phase rotation in frequency domain so that we will follow this approach. Assuming that the channel is fully covered by the cyclic prefix even after misplacement of the FFT window, the effect of a fractional time shift Δt of the FFT window with respect to the initial position is a phase rotation of each symbol S_k in subcarrier k according to

$$\widetilde{S}_k = S_k \cdot e^{j2\pi(\Delta t \cdot f_T)k/N_{\mathrm{DFT}}} = S_k \cdot e^{j2\pi\Delta\tau k/N_{\mathrm{DFT}}} \tag{80}$$

$\Delta\tau = (\Delta t \cdot f_T)$ denotes the fractional offset in multiples of samples and f_T the clock (or sampling) frequency. Just like the receiver is able to easily correct the common phase error due to residual CFO and phase noise, it can compensate the mean timing offset $\Delta\tau$. For a constant clock frequency offset Δf_T without severe clock jitter, $\Delta\tau$ is equal to the time offset valid after half of the symbol duration.

The progressing clock drift during the symbol leads to some loss of orthogonality. If $\Delta\tau(n)$ for some considered OFDM symbol n can be perfectly compensated, the degradation due to the clock drift has been shown to be approximately equal to ([T. 94])

$$D(k) \approx 10 \cdot \log_{10}\left(1 + \frac{1}{3}\frac{E_s}{N_0}(\pi k \frac{\Delta f_T}{f_T})^2\right) \tag{81}$$

The corresponding additional noise variance seen at subcarrier k reads

$$\sigma^2_{\text{clock}}(k) \approx \frac{1}{3}E_s\left(\pi k \frac{\Delta f_T}{f_T}\right)^2 \tag{82}$$

This noise variance rises quadratically with the subcarrier index (and frequency), so that larger DFT sizes are more sensitive to clock deviation. Typically, the system components of mass-product devices will be chosen as to ideally balance performance versus cost. For this reason, it is advantageous to have a system which can cope with higher clock frequency offsets.

We may assume a typical crystal oscillator frequency deviation of 25 ppm, so that the worst-case RX/TX clock deviation would be around 50 ppm. For a large DFT size of $N_{\mathrm{DFT}} = 1024$ and a bandwidth utilization of no more than 80%, which roughly corresponds to 830 subcarriers, the degradation for a symbol SNR of E_s/N_0 = 20 dB yields a worst-case value of $D \approx 10\log_{10}(1 + (1/3) \cdot 100 \cdot (\pi \cdot 415 \cdot 20 \cdot 10^{-6})^2 \approx$ 0.6 dB for the utmost subcarriers. The mean degradation D for K active subcarriers can be defined as follows.

$$\sigma^2_{\text{clock}} := \tfrac{1}{K}\textstyle\sum_k \sigma^2_{\text{clock}}(k) \quad D \approx 10\log_{10}\left(1 + \tfrac{\sigma^2_{\text{clock}}}{N_0}\right) \tag{83}$$

For the considered case, $D \approx 0.2$ dB, which can be regarded as acceptable.

4.4 Carrier frequency offset estimation

Carrier frequency offset (CFO) estimation and correction is usually the first step performed in the receiver together with frame detection. Without CFO correction, channel estimation and fine frame timing correction cannot be performed. CFO correction is most often carried out with the aid of a known preamble providing training symbols to facilitate estimation with low complexity ([T. 97], [IEEa]). More precisely, many schemes follow the principle of a periodic training sequence, because its periodicity is preserved after passing a linear multipath channel and can be exploited for coarse frame detection and CFO correction ([T. 97], [Ric00]). Since the preamble has finite length, an initial guard time needs to be reserved for all essential delayed paths, following the same principle as for the data symbols. This means that the beginning of the repeated training sequence is unused. In any case, this or even a longer guard time is required for the automatic gain control (AGC) loop to settle down. Typically, the beginning of the frame is heavily distorted through clipping of the A/D converters until the gain of the variable-gain amplifier (VGA) is sufficiently reduced.

The accuracy of the CFO correction depends on the signal-to-noise ratio and is studied in [P.H94]. On the other hand, we may expect an influence of oscillator phase noise due to the relatively high experienced values for typical 60 GHz integrated oscillators. Both phase noise and residual carrier frequency offset have a detrimental effect on OFDM demodulation performance. But the influence of phase noise on the variance of the residual CFO after correction constitutes a "second-order" effect, which has an additional performance impact.

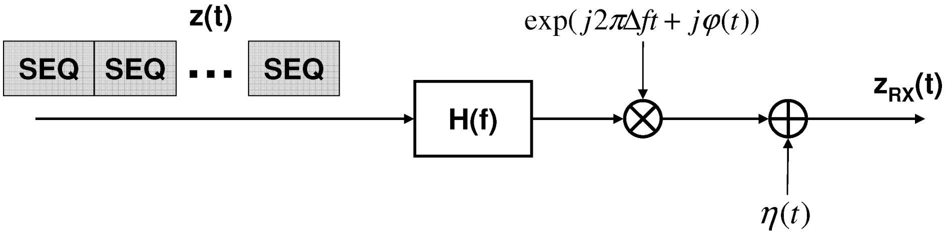


Figure 24: simplified channel model for CFO estimation analysis

In ([M. 07]), the author presented results for a flat channel where only phase noise was considered. In the following, white noise is also included in the analysis to derive optimized system parameters. Simulation is done for AWGN and static frequency selective channels (Figure 24). The transmission signal before and after passing the channel is denoted as $z(t)$ and $z_{CH}(t)$ respectively. The transmit signal will be received with some carrier frequency offset Δf, disturbed by VCO phase noise $\phi(t)$ and AWGN noise $\eta(t)$. If the VCOs on the transmitter side and receiver side can be modeled as Wiener oscillators with diffusivities D_{TX} and D_{RX}, they can be replaced by a single Wiener oscillator with diffusivity $D = D_{TX} + D_{RX}$ at the receiver without falsifying the results.

$$\begin{aligned} z_{CH}(t) &= z(t) \otimes h(\tau) \\ z_{RX}(t) &= z_{CH}(t) \cdot \exp(j(2\pi\Delta f t + \phi(t))) + \eta(t) \end{aligned} \tag{84}$$

The initial corrupted sequence part can be left out of the analysis leaving N_s equal sequences of duration T_s, so that the total length of the uncorrupted part is equal to $T_a = N_s \cdot T_s$. A common estimator is a sort of autocorrelation performed over a finite time interval as given below ([T. 97],[Ric00]):

$$v = \frac{1}{T_I} \int_{t=0}^{T_I} \overline{z_{RX}(t)} z_{RX}(t + T_\Delta) dt \tag{85}$$

T_I indicates the integration interval and T_Δ the step width of the correlator. Apparently, the available integration interval is equal to the total observation interval T_a reduced by the step size T_Δ.

$$T_I = T_a - T_\Delta \tag{86}$$

If no noise components were present, combining (84) and (85) would yield

$$v = P_{z_{RX}} \exp(j2\pi\Delta f T_\Delta) \tag{87}$$

where $P_{z_{RX}}$ denotes the average received power of the sequence. The angle of this complex number is proportional to the frequency offset. The maximum resolvable frequency deviation is equal to

$$|\Delta f_{max}| = 1/(2T_\Delta) \tag{88}$$

In [P.H94], Moose considers two sequences having DFT length. These sequences are compared in frequency domain after DFT operation, so that $T_I = T_\Delta = T_s = T_a/2 = T_{\mathrm{DFT}}$. Despite the transformation, the principle is similar and the estimator can correct frequency deviations up to half a subcarrier spacing. Large frequency offsets cause self noise in form of intercarrier interference.

In Schmidls method ([T. 97]), two identical sequences are related, which have half the size of an OFDM symbol. This happens in time domain using (85). The tolerance range is extended to one subcarrier spacing, at the price of a lower estimation gain. Additional computational effort must be spent to extend the tolerance range, rising the complexity considerably.

For this reason, ten even shorter sequences have been incorporated in the first part of the preamble used in the 802.11a standard [1]. Each short sequence has a length equal to the guard time and extends CFO correction range to two subcarrier spacings.

Troya derives an estimator for this preamble, which attains good estimation performance *and* and offers extended range. Two correlators running in parallel ([Alf04]). The coarse estimator performs an ACF over adjacent short symbols to determine the *integer* frequency offset as multiples of subcarrier spacings. The other correlator relates two long sequences, each composed of four short sequences to estimate the *fractional* frequency offset with a higher accuracy. None of these works consider CFO

---

[1] A second preamble part consisting of two long symbols can also be exploited for CFO correction

estimation accuracy degradation due to phase noise, since phase noise is assumed to be low enough to be neglected.

For the successive analysis, we may assume that the CFO does not exceed the maximum frequency deviation. Each noise contribution will be considered separately setting either $\eta(t) = 0$ or $\phi(t) = 0$, starting with phase noise ($\eta(t) = 0$).

4.4.1 Phase-noise induced frequency error

In Appendix A, the phase noise induced error variance of the CFO estimator is derived analytically (with $\eta(t) = 0$). The error depends on the purity of the Wiener oscillator, expressed by the double-sided linewidth $\Delta f_{\mathrm{3dB}} = D/\pi$ or diffusivity D, and the integration time T_I and step size T_Δ. For $T_I \geq T_\Delta$, the obtained variance is given by

$$\sigma^2_{\Delta f_{\mathrm{err}}} = \frac{2D}{(2\pi)^2} \cdot \frac{3T_a - 4T_\Delta}{3(T_a - T_\Delta)^2} = \frac{\Delta f_{\mathrm{3dB}}}{2\pi} \cdot \frac{3T_a - 4T_\Delta}{3(T_a - T_\Delta)^2} \tag{89}$$

We assume that the step size is equal to the duration of a sequence, $T_\Delta = T_s$. If $N_s = 2$ identical sequences are correlated, then $T_I = T_\Delta = T_a/2$ and we obtain for the standard deviation of the frequency error:

$$N_s = 2 \Rightarrow \sigma_{\Delta f_{\mathrm{err}}} = \frac{1}{2\pi}\sqrt{\frac{4}{3}\frac{2D}{T_a}} \tag{90}$$

An infinite number of sequences leads to the minimum variance, for which we get

$$\sigma_{\Delta f_{\mathrm{err}}} \overset{N_s \to \infty}{\longrightarrow} \frac{1}{2\pi}\sqrt{\frac{2D}{T_a}} \tag{91}$$

The simple result in (91) can be explained as follows. For $N_s \to \infty$ (and, therefore, $T_\Delta \to 0$), the estimator essentially becomes a *phase unwrapper*, which measures the difference of the unwrapped phase at $t = T_a$ and $t = 0$. With Equation (5), the standard deviation of the frequency error is therefore given by $\sigma_{\Delta f_{\mathrm{err}}} = \sigma_\phi(T_a)/(2\pi T_a) = \sqrt{2DT_a}/(2\pi T_a) = \sqrt{2D/T_a}/(2\pi)$. In case of using $N_s = 2$ sequences, the average error increases only by a small factor of $\sqrt{4/3} \approx 115\%$.

The introduced intercarrier interference depends on the ratio of frequency offset to subcarrier spacing, $\gamma = \Delta F/\Delta f$. Therefore, we rewrite (89) and (91) for a discrete time system with sampling frequency $f_T = 1/T$ as follows.

$$\sigma^2_\gamma = \left(\frac{\sigma_{\Delta f_{\mathrm{err}}}}{\Delta f}\right)^2 = \frac{\beta}{2\pi} \cdot \frac{3N_a - 4N_\Delta}{3(N_a - N_\Delta)^2} \cdot N_{\mathrm{DFT}} \tag{92}$$

$$\sigma^2_{\gamma,\min} = \left(\frac{\sigma_{\Delta f_{\mathrm{err,min}}}}{\Delta f}\right)^2 \approx \frac{\beta}{2\pi}\frac{N_{\mathrm{DFT}}}{N_a} \tag{93}$$

$$\Delta f = f_T/N_{\mathrm{DFT}} \quad \gamma = \Delta F/\Delta f \quad \beta = \Delta f_{\mathrm{3dB}}/\Delta f \quad N_a = T_a \cdot f_T \quad N_\Delta = T_\Delta \cdot f_T$$

The equations have been validated by simulation, as shown in Figure 25, for $N_{\mathrm{DFT}} = 1024$. The considered cases are:

1. An estimator with a fixed step size of N_Δ = 256.

2. An estimator, which correlates two symbols, $N_\Delta = N_a/2$.

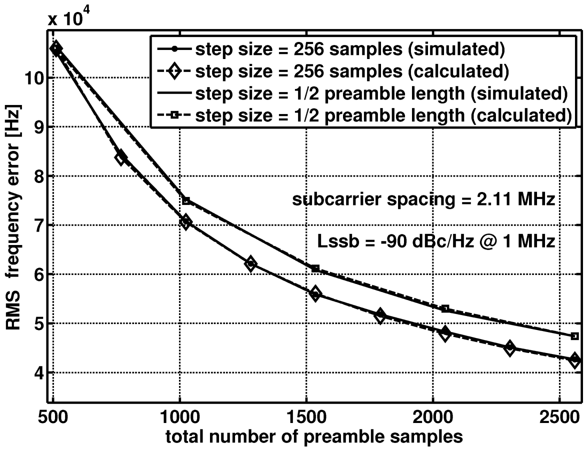


Figure 25: RMS frequency estimation error due to Wiener phase noise

4.4.2 AWGN noise induced frequency error

Now the AWGN induced frequency error is considered ($\phi(t) = 0$). The AWGN noise will be bandlimited by any filter in the receiver RF path, including the anti-aliasing filter. In order to avoid dealing with statistics of bandlimited noise, analysis can be performed in the discrete time domain after optional sampling rate reduction. In this case, the signal bandwidth will typically occupy about 80-90% of the Nyquist bandwidth $f_T/2$. For simplicity, we may assume uncorrelated AWGN noise after passing an ideal Nyquist filter. With the definitions $T = 1/f_T$, $N_I = T_I/T$, $N_\Delta = T_\Delta/T$, we obtain for corresponding *discrete-time estimator* the following expressions.

$$
\begin{aligned}
v &= \frac{1}{N_I}\sum_{n=0}^{N_I-1}\overline{z_{\text{RX}}(nT)}z_{\text{RX}}(nT+N_\Delta T) = \frac{1}{N_I}\sum_{n=0}^{N_I-1}\left[\overline{z_{\text{CH}}(nT)}\exp(-j2\pi\Delta f T n)+\overline{\eta(nT)}\right]\cdot \\
&\left[z_{\text{CH}}(nT+N_\Delta T)\exp(j2\pi\Delta f T(n+N_\Delta))+\eta(nT+N_\Delta T)\right] \\
&= \exp(j\underbrace{2\pi\Delta f T N_\Delta}_{\phi_\Delta})\cdot\underbrace{\frac{1}{N_I}\left[\sum_{n=0}^{N_I-1}|z_{\text{CH}}(nT)|^2\right]}_{P_{z_{RX}}}+\underbrace{\frac{1}{N_I}\left[\sum_{n=0}^{N_I-1}\overline{\eta(nT)}\eta(nT+N_\Delta T)\right]}_{B} \\
&+\underbrace{\frac{1}{N_I}\left[\sum_{n=0}^{N_I-1}\overline{\eta(nT)}z_{\text{CH}}(nT+N_\Delta T)\exp(j2\pi\Delta f T(n+N_\Delta))\right]}_{C_1} \\
&+\underbrace{\frac{1}{N_I}\left[\eta(nT+N_\Delta T)\overline{z_{\text{CH}}(nT)}\exp(-j2\pi\Delta f nT)\right]}_{C_2}
\end{aligned}
\tag{94}
$$

$$
\begin{aligned}
\Leftrightarrow v &= P_{z_{RX}}\exp(j\phi_\Delta)+B+C_1+C_2 \\
&= P_{z_{RX}}\exp(j\phi_\Delta)\left[1+B\exp(-j\phi_\Delta)/P_{z_{RX}}+C_1\exp(-j\phi_\Delta)/P_{z_{RX}}+C_2\exp(-j\phi_\Delta)/P_{z_{RX}}\right]
\end{aligned}
\tag{95}
$$

The frequency offset is estimated to

$$\begin{aligned}\widehat{\Delta f} =& \frac{1}{2\pi T N_\Delta}\angle(v) \qquad (96)\\ =& \Delta f + \frac{1}{2\pi T N_\Delta}\angle\left[1 + B\exp(-j\phi_\Delta)/P_{z_{RX}} + C_1\exp(-j\phi_\Delta)/P_{z_{RX}} + C_2\exp(-j\phi_\Delta)/P_{z_{RX}}\right]\end{aligned}$$

$P_{z_{RX}}$ is identified as the signal power seen at the receiver. For high SNR, we may neglect term B. Also, even for moderate SNR, all complex noise terms in Equation (96) have amplitudes much smaller than unity so that (96) can be well approximated to

$$\widehat{\Delta f} \approx \Delta f + \frac{1}{2\pi T N_\Delta P_{z_{RX}}}\mathrm{Im}\{C_1\exp(-j\phi_\Delta) + C_2\exp(-j\phi_\Delta)\} = \Delta f + \Delta f_{\mathrm{err}} \qquad (97)$$

The case, which is simple to evaluate, is given for $N_\Delta \geq N_I$, since every noise contribution $\eta(nT)$ in (97) appears only once. The complex noise power is $E\{\eta(nT)^2\} = N_0$. Since there are $2N_I$ independent error terms, the total error variance is given to

$$\begin{aligned}\sigma^2_{\Delta f_{\mathrm{err}}} &= \frac{1}{(2\pi\cdot T\cdot N_\Delta\cdot P_{z_{RX}})^2}\cdot\frac{2N_I\cdot(N_0/2)\cdot P_{z_{RX}}}{N_I^2} = \frac{f_T^2}{(2\pi)^2}\cdot\frac{1}{N_\Delta^2\cdot N_I\cdot(P_{z_{RX}}/N_0)} \qquad (98)\\ &= \frac{f_T^2}{(2\pi)^2}\cdot\frac{1}{N_\Delta^2\cdot N_I\cdot\mathrm{SNR}} = \frac{f_T^2}{(2\pi)^2}\cdot\frac{1}{N_\Delta^2\cdot(N_a - N_\Delta)\cdot\mathrm{SNR}}\end{aligned}$$

If two regular OFDM symbols are compared, then $N_I = N_\Delta = N_{\mathrm{FFT}}$, and we obtain the same error variance as in [P.H94]. [T. 97] claims that this estimator attains the Cramer-Rao lower bound. It seems that a better estimate is still possible if the signal amplitudes $|z_{\mathrm{CH}}(nT)|$ can be estimated and exploited through maximum ratio combining. Also, it is easy to show that (98) is minimized for $N_\Delta = (2/3)N_a$ instead of $N_\Delta = N_a/2$ [1].

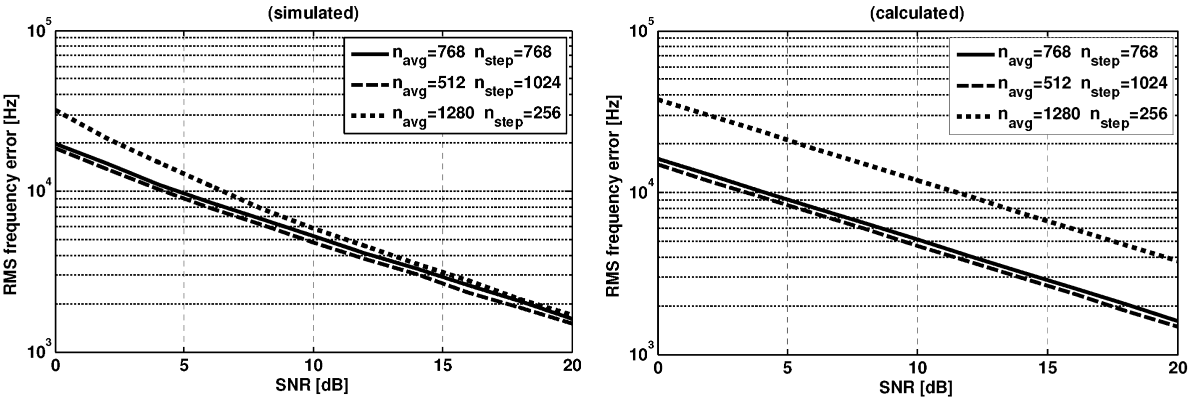


Figure 26: Simulated vs. calculated carrier frequency estimation error in AWGN channel

Simulation of the RMS frequency error versus its calculation is shown in Figure 26 for a sampling rate of f_T = 2.16 GHz and three different parameter sets (N_I, N_Δ) for the estimator ($N_{\mathrm{avg}} \equiv N_I$,

---

[1] It appears that the CFO range is reduced, but running two correlators parallel, one with $N_\Delta = (2/3)N_a$ and the other with $N_\Delta = N_a/2$, and selecting the first one, when the estimated values are about the same, circumvents this limitation.

$N_{\mathrm{step}} \equiv N_\Delta$). As predicted, choosing $N_\Delta = (2/3)N_a$ gives slightly better performance. For the first and second set, the calculated error is accurate starting from a SNR of about 6 dB. At very low values, equation (98) is too optimistic, since the neglected noise product term $\overline{\eta(nT)}\eta(nT + N_\Delta T)$ contributes to the CFO error.

The third estimator just compares adjacent short sequences, so that $N_\Delta < N_I$. Equation (98) strongly overestimates the CFO error and cannot be used. Each noise value $\eta(nT)$ appears twice in (97), and some noise cancellation effect is observed. The difference in noise performance compared to the other estimators becomes smaller at higher SNR values. An exact analytical expression for the CFO error for $N_\Delta < N_I$ has not been found.

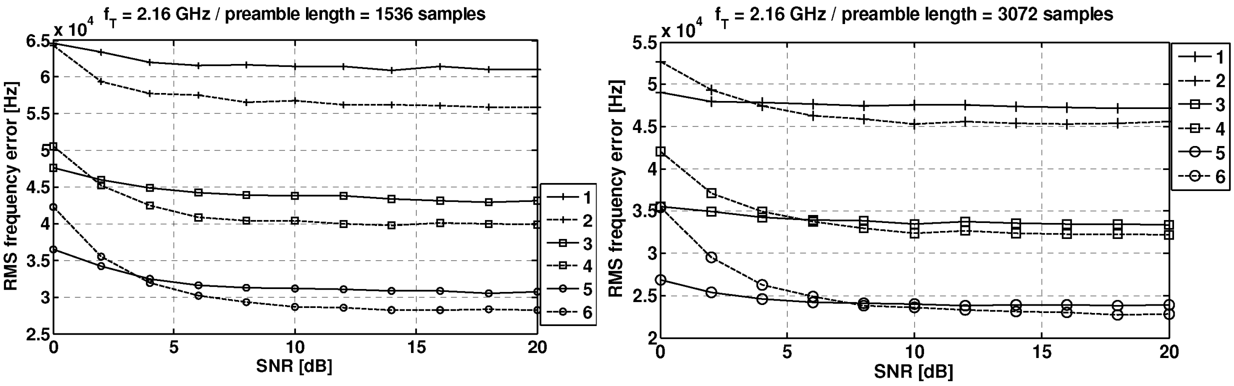


Figure 27: CFO error for phase noise and AWGN.

1,2: L_{ssb} = -87 dBc/Hz; 3,4: L_{ssb} = -90 dBc/Hz; 5,6: L_{ssb} = -93 dBc/Hz, all values at 1 MHz offset from the carrier. Solid curves: N_Δ=768 estimator; dashed curves: N_Δ=256 estimator

Finally, we consider the combined effects of white noise and phase noise in Figure 27 for preamble lengths of 1536 and 3072 samples at 2.16 GHz sampling frequency. For realistic phase noise values in the range of -87 dBc/Hz up to -93 dBc/Hz at 1 MHz offset, we observe that CFO estimation accuracy is dominated by phase noise for SNR values above 5 dB in more or less all cases. The small-step estimator is always preferable for moderate to high SNR values. We conclude that a considerably higher preamble length must be provided in the frame due to the high phase noise.

4.4.3 Probability to exceed a given absolute frequency error

An overview of the synchronization scheme for the receiver is given in section 5.7. The receiver relies on the initial CFO estimate obtained with the preamble. This estimate is used for the entire data frame. As previously discussed, the residual CFO error shows up as a correctable common phase error and ICI for each OFDM symbol. The resulting SNR degradation of the system caused by this ICI was given with Equation (70) in Section 4.3.2. It is reproduced here for convenience.

$$D_{dB} \approx \frac{10}{3 \cdot \ln 10} \cdot \left(\pi \frac{\Delta F}{\Delta f}\right)^2 \cdot \frac{E_s}{N_0} \tag{99}$$

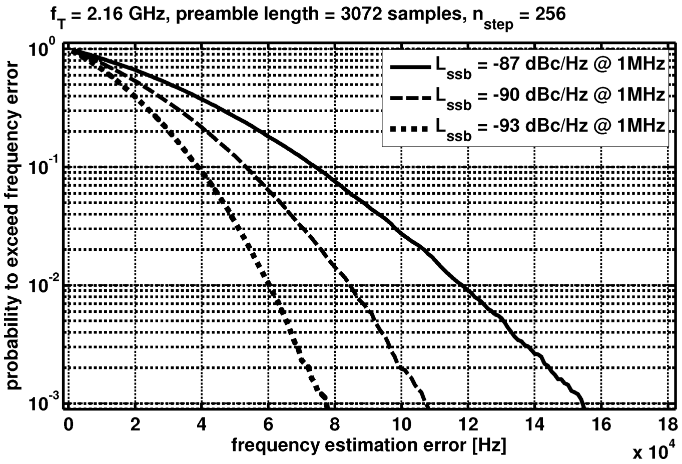


Figure 28: Outage probability for frequency error

The degradation becomes more significant for higher symbol SNR values for which we may want to use higher data modes. We have seen that for moderate to high SNR values, the frequency error is mainly driven by phase noise of integrated VCOs in current Si-Ge technology, and the influence of white noise may be neglected.

So far we have focused on the RMS frequency error as an average over a large number of frames. We may also consider the distribution of the frequency error and ask for the probability to exceed a certain absolute error, which can be associated with some SNR degradation. For this matter, the distribution of the frequency estimation error for Wiener phase noise was obtained by simulation. It is shown in Figure 28 for three different phase noise levels.

For example, we may assume a high symbol SNR of E_s/N_0=100 (or 20 dB) prior to degradation and a subcarrier spacing of $\Delta f = 2$ MHz. Then, according to this figure and Equation (99), a phase noise level of -93 dBc/Hz at 1 MHz offset (seen by the receiver) would yield an SNR degradation of more than approximately 0.5 dB for only 10% and 1 dB for 1% of the transmitted frames. This may be regarded as an acceptable performance penalty. A better quantification is provided with system simulations in section 5.

5 PHY layer specification and performance investigation

5.1 Overview and general considerations

The design of an OFDM based physical layer can be split into a number of specification tasks performed in a certain order. Prior to this work one can think about the incorporated features of the target system and trade system performance against complexity. In addition, the system should be designed in awareness of the considered receiver algorithms and with a realistic channel model. The investigation of PHY layer parameters and algorithms in this work has been performed in an iterative way. Therefore, a clear road map of specification tasks was not followed.

In order to restrict the various possibilities, it was decided to use bit-interleaved coded modulation (BICM) with OFDM transmission. We follow the WLAN standard 802.11a and use *coherent* BICM with BPSK, QPSK and higher order QAM-modulation in the subcarriers. We assume that sufficiently accurate synchronization and channel estimation can be performed with little pilot overhead. An interesting alternative would have been the application of a differential scheme like Multilevel Differential Modulation ([Her99]).

For its simplicity, a convolutional code has been used as the primary channel code. A brief recapitulation of convolutional codes is given in Appendix D. Additional coding gain can be achieved with an outer Reed Solomon code. As an alternative, an LDPC code with rather moderate block size has been applied in order to keep the complexity low. Better code performance would be achieved with a well chosen Turbo code ([C. 96]), but decoder implementation is more involved.

OFDM with convolutional encoding requires interleaving to spread adjacent code bits across the used spectrum. In this way, frequency diversity is exploited to reduce the probability that adjacent code bits are suffering from a deep fade. In the IEEE 802.11a standard, there is fixed block interleaving pattern assigned to each particular mode of modulation. From the theoretical point of view, this approach can already be regarded as a compromise, since the channel capacity in case of frequency selective fading is not optimally exploited.

Bit-loading is an advanced technique which aims to determine the number of mapped code bits for each particular subcarrier according to the estimated subcarrier SNR. This requires channel state information at the transmitter to be provided by the receiving station. On the other hand, the receiver must know the subcarrier load configuration for every data frame. This may lead to a significant signaling overhead in case of short data frames. In addition, adaptive interleaving is required to account for the efficiency of the channel code. More frequent signaling is required when the channel becomes time-variant. Although a promising technique, bit loading has not been considered in this work due to the additional complexity associated with it and the need for a return link.

The next issue to discuss is the insertion of pilot information for the aid of initial synchronization and channel estimation and for tracking of time-variant link parameters during the frame. The first task is usually done with the aid of a preamble preceding the data frame. For tracking, different solutions have been applied. For Digital Video Broadcasting Terrestrial (DVBT), so-called scattered pilots are sparsely located within the subcarrier time-frequency grid to allow the receiver to interpolate the time-varying channel coefficients in both time and frequency. The rate of inserted pilots is set as to fulfill the sampling theorem on both axes. In contrast, pilots for the aid of channel tracking are omitted in the 802.1a WLAN standard. Instead, data-aided decision feedback equalization (DFE) is applied. This scheme uses the processed (equalized, demapped, deinterleaved and decoded) data bits to reconstruct the ideal subcarrier symbols in a feedback loop (applying re-encoding, re-interleaving and re-mapping). The reconstructed "ideal" symbols are set in relation to the noisy symbols originally received to perform a slow adaptation of the channel coefficients. Decision-feedback can provoke error propagation after a wrong bit decision. Since a purely packet based communication is primarily used for the exchange of data, a single source bit error will cause retransmission of the frame. Therefore, the frame error rate and not the bit error rate is determining the performance and error propagation is of minor importance.

For the considered system, we may not be able to implement DFE as efficiently as usual because the high receiver latency will impose limitations on the adaptation process. In many references, OFDM symbol data is already incorporated in a channel update for the next symbol. In contrast, each update might be applied only on OFDM symbols coming 10 to 20 slots later. Channel interpolation also requires a significant additional hardware effort. If pilots are interpolated both in time and frequency, a larger amount of symbols need to be buffered, and the receiver latency is significantly increased.

Because of the high latency of the system and for the sake of simplicity, two choices seem most appropriate. Firstly, the frame may be limited to a duration shorter than the coherence time of the channel. Secondly, we may embed additional channel re-estimation symbols in the frame. If such midamble symbols are chosen identical to the symbols in the preamble, the same hardware blocks can be re-used to process them. Compared to the pilot-aided interpolation scheme this approach gives inferior performance, because it follows the sample-and-hold principle. On the other hand, the insertion of midambles and their rate of appearance can be decided according to the channel conditions. Furthermore, it may be assumed that domestic and office indoor communication will face quasi-static radio channels most of the time.

Even if no channel interpolation will be performed, sparsely inserted continuous pilot subcarriers are often used for phase correction. Phase errors are caused by a residual carrier frequency error, phase noise and clock drift. These effects lead to some partial loss of orthogonality, seen as intercarrier interference, and to phase errors in the subcarriers. At least, the phase errors can be corrected with the pilots.

Midamble insertion, when needed, enables the transmission of long frames. Long frames are efficient when combined with MAC packet aggregation. In general, frame error rate rises with the packet length. As an example, we can take uncoded modulation. For a bit error rate of P_b and packet length N_p, packet error rate is equal to $P_P = 1 - (1 - P_b)^n \approx n \cdot P_b$ if P_b is orders of magnitude smaller than unity. To lower the frame error rate, a physical frame may consist of several MAC packets. This enables the transmitter to resend only the corrupted MAC packets and fill the rest of the PHY frame with the next data packets. For a fixed maximum ratio between preamble length and data frame length, longer frame durations also allow the use of a longer preamble in order to obtain better estimates of synchronization parameters and/or channel coefficients. Since midamble-based channel re-estimation does not use decision-feedback, no error propagation can arise.

Investigation of the coded OFDM modulation scheme needs to consider (antenna dependent) channel characteristics in terms of channel delay spread, coherence bandwidth and coherence time as well as RF impairments. Interference created by RF impairments and residual ISI leads to some performance degradation. In the simplest form, this degradation is expressed as a loss of the average subcarrier signal-to-noise ratio. Incorporated pilot data for synchronization, channel estimation and tracking constitutes additional overhead. The same is true for the cyclic prefix (or suffix). Moreover, implementation losses arise when parameters are not perfectly estimated. Ideally, the target should be to specify OFDM system parameters in such a way as to minimize the overall performance degradation. But this can only be done if all involved receiver algorithms and their dependencies on system parameters are known in advance. Depending on the application and allowed system complexity, different optimization criterions could be applied.

- According to 802.15.3c (TG3c) selection criteria ([IEE07]), video transmission without retransmission calls for a bit error rate after decoding to be lower than 10^{-6}.
- For packet based data transmission with optional retransmission, the TG3c group has specified a maximum tolerable frame error rate of 8%.
- For a pure point-to-point demonstration link without other stations involved, we might simply aim to maximize the goodput (effective source data throughput) R_G.

For the moment, we may follow the third criterion. The goodput R_G is given as

$$\begin{aligned} R_G &= f_T \cdot \frac{N_m}{N_{\text{frame}}} \cdot (1 - \text{FER}) = f_T \cdot \frac{N_m}{N_{\text{df}}} \cdot \gamma_{\text{pre}} \cdot (1 - \text{FER}) \\ \gamma_{\text{pre}} &= \frac{N_{\text{df}}}{N_{\text{frame}}} = \frac{1}{1 + \frac{N_{\text{pre}}}{N_{\text{df}}}} \end{aligned} \tag{100}$$

where $N_{\text{frame}} = N_{\text{df}} + N_{\text{pre}}$ denotes the total frame duration in samples, N_{df} and N_{pre} the duration of data part and preamble part, γ_{pre} stands for the loss factor caused by the preamble overhead, N_m denotes the number of message bits, and $f_T = 1/T$ the sampling frequency. Note that the preamble loss γ_{pre} depends on the payload size. The PHY data rate is approximately equal to $R_{\text{phy}} \approx f_T \frac{N_m}{N_{\text{df}}}$. This is the netto data rate without preamble overhead. It differs from the real data rate only in the sense that additional dummy bits carrying no information need to be appended to fill the last code block and last OFDM symbol in order to obtain a valid frame structure. Assuming a fixed constellation mapping of m_{sc} bits for every data subcarrier, a code rate of $r < 1$, N_{dsc} active data subcarriers, an DFT size of N_{FFT} and a guard duration of N_g samples (appearing as prefix or suffix), the PHY rate is given to

$$\begin{aligned} R_{\text{phy}} &= f_T \cdot \frac{N_{dsc} \cdot m_{sc} \cdot r}{N_{\text{FFT}} + N_g} = f_T \cdot \frac{N_{dsc} \cdot m_{sc} \cdot r}{N_{\text{FFT}}} \cdot \gamma_g \\ \gamma_g &= \frac{1}{1 + \frac{N_g}{N_{\text{FFT}}}} \end{aligned} \tag{101}$$

γ_g is the loss in data rate caused by the guard time. When considering continuous pilot subcarriers, we can think of a maximum data rate using the signal bandwidth without any pilots. Then the loss of data rate comes in form of an exchange of a certain amount of data subcarriers into pilots. For a total number of subcarriers $N_{sc} = N_{dsc} + N_p$, N_p denoting the number of pilots and γ_p the pilot loss, we write

$$R_{\text{phy}} = f_T \cdot \frac{N_{dsc} \cdot m_{sc} \cdot r}{N_{\text{FFT}}} \cdot \gamma_g = f_T \cdot \frac{N_{sc} \cdot m_{sc} \cdot r}{N_{\text{FFT}}} \cdot \gamma_g \cdot \gamma_p \tag{102}$$

$$\gamma_p = 1 - \frac{N_p}{N_{sc}}$$

Combining the last three equations yields

$$R_G = f_T \cdot \frac{N_{sc} \cdot m_{sc} \cdot r}{N_{\text{FFT}}} \cdot \gamma_{\text{pre}} \cdot \gamma_g \cdot \gamma_p \cdot (1 - \text{FER}) \tag{103}$$

We assume that the preamble is transmitted with the same power level as the data frame. The same is true for the prefix/suffix. Finally, we assume that all subcarriers are transmitted with equal power. Every exchange of a data subcarrier into a pilot subcarrier is a loss of data rate and also a loss of power (and energy), which could otherwise be used for the remaining data subcarriers to rise the SNR, in case that this pilot was not transmitted. Then in all three cases data rate loss is equal to energy loss. For the rest of the work, BER/FER performance is always related to the Eb/N0 ratio, given by Equation (49), which accounts for all fixed and performance-dependent losses.

The frame error rate in (103) is the outcome of the performance of the complete system with the chosen parameters for a given SNR, channel scenario and radio front end specification. In addition, it depends on the chosen data mode and amount of message data. TG3c specified a standard payload size of 2048 bytes. Two approaches seem possible.

- The system may be optimized for the standard payload size of 2048 bytes.
- Alternatively, *frame aggregation* may be incorporated to extend the payload size and in this way allow a larger preamble. This may impose some restrictions for the MAC layer.

In the following chapters, the OFDM PHY specifications are progressively elaborated for a narrowband and a wideband OFDM layer, which have been developed and simulated during the WIGWAM and EASY-A project. The text omits the original chronological order of development in favour of a better presentation.

5.2 Investigation of basic OFDM modulation parameters

A rigorous treatment could be based on maximization of the effective throughput as given in (103). But such a strict mathematical approach may not be adequate. The complete knowledge of receiver algorithms would be required in advance. Also, solid information about channel characteristics is needed. Whereas the properties of the 60 GHz radio channel remain unchanged, RF components may likely improve over the years. A system optimized for the existing benchmark components may likely look different a few years later. The decisions on system parameters are sketched below.

5.2.1 Signal bandwidth

For both OFDM demonstrators, parameter selection started from a given signal bandwidth. The 400 MHz Nyquist bandwidth of the first OFDM demonstrator was set by limitations of the processing hardware, as described in the hardware Section 6 and Appendix I. For the wideband system, a Nyquist bandwidth equal to the TG3c channel spacing of 2.16 GHz was chosen.

5.2.2 DFT size, guard length

The DFT size is limited in both directions by different factors. A larger DFT size rises the degradation associated with phase noise (Section 4.3.2). An under-dimensioned DFT size comes either at the price of a higher guard time loss, for a fixed guard time, or rises the ISI for a fixed symbol-to-guard-time ratio (section 4.3.1)[1]. To avoid a loss of more than 1 dB, the guard time N_g is allowed to have a maximum duration of 20% of the complete OFDM symbol of length $N_g + N_{FFT}$.

The required guard time can be effectively reduced by application of a high-gain antenna. Such an antenna also improves the link budget. For a conventional fixed-beam antenna (horn or Vivaldi type), the high gain is paid with a narrow half-power beam width. The receiver, being equipped with such an antenna, needs to point in direction of the transmitter or a strong first-order reflection. This reduces the mobility and may lead to high shadowing loss when an obstacle is present. Here lies the benefit of a patch antenna array, where the beam direction can be controlled by the baseband using some beam search algorithm. But for both demonstrators, a patch antenna array was unavailable. Therefore, a fixed beam antenna had to be assumed. A good compromise between mobility and antenna gain is found for practical half-power beam widths in the order of 30 to 60 degree.

We first consider the wideband system. The channel interference distribution for a guard duration of 256 samples (T_g=118.5 ns) and a DFT size of $N_{FFT} = 1024$ (T_{FFT}=474.1 ns) has been calculated for 1000 channels per model with the aid of Equation (65) and standard histogram techniques. Figure 29 presents the result for various TG3c channels and receiver antenna beam widths between 30° and 60° (table 6 in Section 3.3). CM3.1 and CM3.2 have (almost unrealistically) high delay spread, and therefore degrade performance for higher data modes. Following the rules of TG3c ([IEE07]), the worst 10% of the channels can be discarded. After neglecting the CM3.1 and CM3.2 channels having the highest delay spread, there still remain some channels, which limit the subcarrier SNR to 20 dB or even less.

On the other hand, for the typical phase noise values there is hardly any room to go for a higher DFT size. The degradation for these parameters has already been shown by Figure 21 in Section 4.3.2. A phase noise value of -95 dBc/Hz at 1 MHz is required for each station to limit the degradation to 0.8 dB at 20 dB SNR.

---

[1] Some literature characterizes a channel (model) by a *maximum delay spread* parameter. The author believes that there is no clear definition for this parameter. For any reasonable (not largely oversized) guard time, there will be some remaining ISI created by the tail of the channel impulse response. But the maximum tolerable ISI depends on the transmission mode and, therefore, on the application.

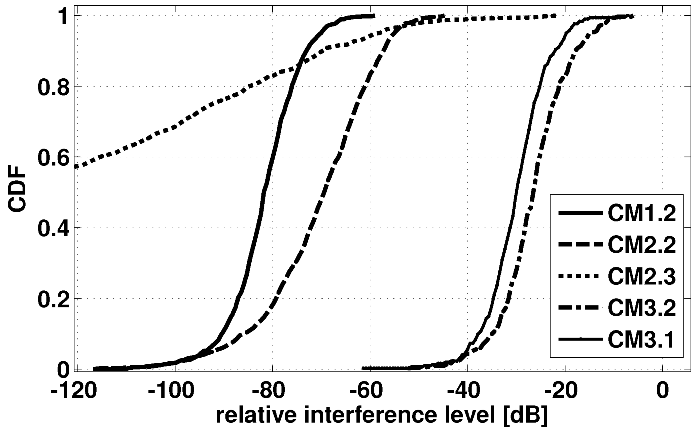


Figure 29: Interference statistics for $T_g = 118.5$ ns and $T_{FFT} = 474.1$ ns

For suitable implementation, the DFT size is usually chosen as a power of two, 2^K. Doubling the DFT is quasi prohibited by phase noise. It is shown in Section 4.4 that the residual carrier frequency offset error is mainly determined by the high phase noise and not the additive white noise. The minimum variance of the normalized frequency error $\gamma = \Delta F/\Delta f$, where Δf stands for the subcarrier spacing, is reproduced here for convenience.

$$\sigma^2_{\gamma,\min} \approx \frac{\beta}{2\pi}\frac{N_{\mathrm{DFT}}}{N_a} \tag{104}$$

$\beta = \Delta f_{3\mathrm{dB}}/\Delta f$ denotes the normalized oscillator line width. For a fixed signal bandwidth $B = N_{\mathrm{DFT}} \cdot \Delta f$, we rewrite (104) as

$$\sigma^2_{\gamma,\min} \approx \frac{1}{2\pi}\frac{\Delta f_{3\mathrm{dB}}}{B}\frac{N^2_{\mathrm{DFT}}}{N_a} \tag{105}$$

Doubling the DFT size, i.e. reducing the subcarrier spacing to half of its value is paid with a four times larger preamble length, if the CFO error variance shall remain constant. Hence, a system with a narrow subcarrier spacing becomes quickly inefficient for short frame transmission, and in fact, the TG3c task group has decided on a wider spacing using a 512-point FFT, heavily relying on high antenna directivity. For this work, a 256-point DFT with Δf=1.5625 MHz has been chosen for the narrowband system and a 1024-wide DFT implying Δf=2.11 MHz for the wideband system. For both cases, the guard duration was chosen to $N_{\mathrm{DFT}}/4$ (1 dB performance penalty), resulting in 160 ns guard time for the narrowband and 118.5 ns for the wideband system.

Finally, we note that a system with a larger guard time can tolerate higher timing synchronization errors. This in turn allows a simpler algorithm with less hardware cost to be used for synchronization (see Section 5.8).

5.2.3 Data, pilot and guard subcarriers

The signal bandwidth of the system is mainly given by the amount of data, pilot and DC zero subcarriers. As the signal bandwidth approaches the Nyquist bandwidth, the requirements for the analogue reconstruction filter or digital interpolation filter become very demanding. A feasible ratio for signal bandwidth to Nyquist bandwidth is around 80-90%.

For the number of pilot subcarriers, we may simply follow the 802.11a standard and specify coarsely the same relative amount of pilots. 802.11a uses 4 pilots and 48 data subcarriers, which is $1/13$ or 7.7% of all subcarriers. For a simpler implementation, the distance of adjacent pilots was maintained to be constant. System simulations will show that the number of pilots is properly chosen.

A DC blocking filter is usually employed to reject large DC offsets. The largest CFO in this system was chosen to two subcarrier spacings. To avoid DC blocking of the subcarrier with the lowest frequency, the number of virtual (unused) DC subcarriers was chosen to 5.

Combining these considerations, the system parameters given in Appendix G were derived. The WIGWAM parameters are obtained by upscaling of IEEE 802.11a parameters. The signal bandwidth is around 333 MHz and 1757 MHz, respectively, which constitutes a fraction of 82% and 81%, respectively. The 16 pilots for the narrowband and 60 pilots for the wideband mode correspond nearly to the same relative amount of pilots as for the IEEE802.11a standard.

For the WIGWAM PHY, a larger bandwidth could have been chosen, since 2x oversampling is used and digital decimation/interpolation with a narrow transition region is easier to accomplish. For the EASY-A PHY, no oversampling can be maintained and reconstruction filtering in the analogue domain becomes harder.

An additional restriction has not been mentioned yet, which also determined the parameter selection. The system features a data granularity of 8 bits = 1 byte. To keep the implementation simple, the number of subcarriers is dividable by 16 for BPSK at a code rate of r=1/2 and also dividable by 3 to support puncturing modes. In this way, the data stored in an OFDM symbol for every PHY mode is an integer multiple of message bytes. With this restriction, the number of subcarriers would have to be a multiple of 48, and for this reason, 192 data subcarriers were chosen. For the EASY-A PHY, the data granularity is increased to four bytes. The chosen number of 768 subcarriers is dividable by $3 \cdot 32 = 96$. In addition, the OFDM block size matches the code block size of the WiMAX LDPC code (768,384).

5.2.4 Pulse waveform and signal spectrum

A non-rectangular pulse waveform has been utilized to reduce out-of-band emissions of the OFDM spectrum. The commonly used continuous-time raised-cosine pulse $W(t)$ is given in Equation (269) in Appendix C.2. The approximated discrete-time pulse has been obtained by sampling. The number of overlapping samples between adjacent OFDM symbols has been chosen to $N_{ov} = 1$ for the narrowband and $N_{ov} = 8$ for the wideband mode. The guard time is only slightly reduced.

The resulting signal spectrum for the narrowband mode is shown in Figure 30. This is the spectrum obtained after the interpolation filter implemented in hardware. The frequency is normalized to 400 MHz, the Nyquist frequency after interpolation. For $N_{ov} = 1$ there is almost no windowing involved. Nonetheless, the filter reduces out-of-band emission by at least 60 dB within the neighbor channel, if the spacing between adjacent frequency channels is chosen to 400 MHz. In this case, the center frequencies of the adjacent channels would be located at a normalized frequency of ± 1.

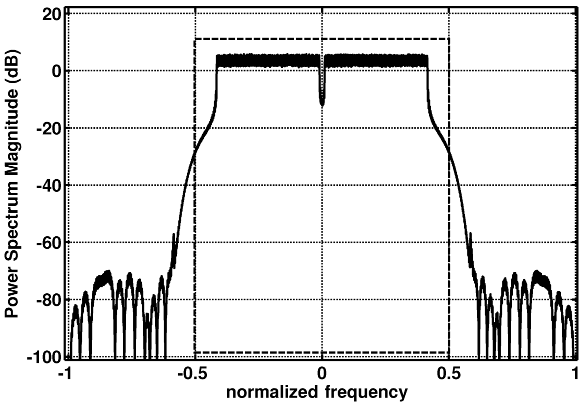


Figure 30: Signal spectrum of narrowband PHY (WIGWAM)

The wideband signal spectrum with and without windowing is shown in Figure 31, virtually using the same interpolation filter [1]. With a channel spacing of 2160 MHz, the adjacent carrier frequencies are again located at ±1. The neighbor spectrum begins at ±0.6 (or ±1300 MHz). With windowing, subcarriers located at the band edge will pick up less interference. Also, there will be less interference created into foreign service bands.

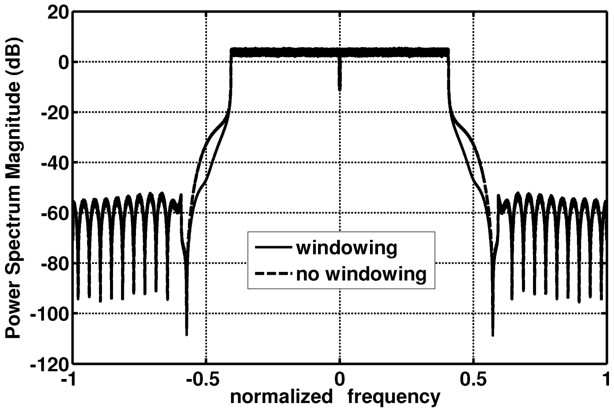


Figure 31: Signal spectrum of wideband PHY (EASY-A) with and without windowing

Note that out-of band interference is actually likely to be dominated by spectral regrowth created by power amplifier nonlinearity ([Beh98]) when the PA is driven slightly into saturation in order to achieve an overall better link budget.

5.3 Selection of a channel code

As pointed out in Chapter 4.1, the inclusion of channel coding in an OFDM system is of fundamental importance, since system performance in frequency selective channels would otherwise be strongly degraded. Furthermore, the difference in coding gain between soft- and hard-decision decoding is

---

[1] Due to hardware limitations, the interpolation filter for the wideband system could not be implemented in the digital domain, rising the requirements for the analog reconstruction filter.

much higher than for single-carrier systems, if the reliability of each subcarrier is incorporated in the related soft bits ([Fil02]). Channel coding constitutes only a fraction of this work, and the intention was not to find the most powerful code within all possible code classes. Instead, the work confines to the selection of existing codes, which have been successfully applied in other radio standards.

To be more specific, the standard convolutional code (133,171) used in IEEE 802.11a can be chosen for its simplicity and efficiency as the basic channel code ([IEEa]). According to [Ode76], this is the best known code with code rate R = 1/2 for a constraint length of K=7. This constraint length value represents a good trade-off between error correction capability and decoding complexity. The achieved *minimum free distance* for this code is $d_{free} = 10$. The free distance represents the minimum number of code bits, in which any two code words will differ and mainly determines the bit error rate.

Convolutional codes can be regarded as inferior to Turbo codes, but there is a way to enhance performance by application of an outer, hard-decision based Reed-Solomon code. The "industrial standard" RS code (239,255) operates on bytes and is able to correct any arbitrary eight out of 255 code bytes. This is paid with a slight decrease of the total code rate by a factor of $R_{\mathrm{RS}} = 239/255 \approx 0.937$. The advantages of this concatenated code scheme are well known ([Ber94]). Bit errors caused by a wrong path decision of the Viterbi decoder appear often in bursts and they can be efficiently corrected by the RS decoder. The RS code offers good burst error performance, since the number of erroneous bits in a wrong byte does not count.

A short LDPC code has also been considered as an alternative solution. The (768,384) LDPC code taken from the WiMAX standard has been investigated at a later stage of the project. A performance comparison is given in Section 5.5.4.

5.4 Stream arrangement and frame structure

In order to achieve a data rate greatly exceeding the rate processable by a single convolutional decoder (see Section 6.10), it is necessary to divide the message stream into several substreams, which are encoded and decoded separately.

The code streams must be somehow arranged in the OFDM frame. One way is to divide the available bit positions in an OFDM symbol among the code streams (solution A). Each subcarrier may be assigned to just one code stream (like in OFDMA) or shared by different streams. The latter is possible for modulations higher than BPSK, where different bit positions in each subcarrier may be assigned to different streams.

Another possibility is to dedicate each OFDM symbol to one stream only and apply a form of TDMA to divide the frame among the streams (solution B). Without claiming that TDMA-based mapping gives superior performance, it had been decided at an early stage of the project to follow this approach for several practical reasons.

- If the channel is static and if the link quality does not continuously degrade over time, a TDMA-based code stream mapping should give about the same performance as if only one code stream would have been used.

- It is possible to specify the number of simultaneous convolutional code streams to be a variable parameter. Transmitter and receiver can decide on the number of streams using a handshake protocol. If both stations operate at a high clock speed using dedicated hardware, the number of streams can be reduced, if this leads to any significant advantage.
- For solution B, the interleaving scheme is independent of the stream number. In contrast, the number of code streams must be fixed for solution A. Otherwise, the interleaving scheme needs to be adapted to support a variable stream number for the same modulation.

A frame arrangement for four streams is exemplarily shown in Figure 32, including a possible data flow in the receiver. Each code processing block consists of a low-rate de-interleaver and decoder. Demultiplexing of message data at the transmitter and multiplexing of (hopefully) the same data at the receiver can be handled in bursts. This means that the receiver always collects the full data packet of one received OFDM symbol, before switching to the next stream. This may require some small buffer stage before and after each de-interleaver/decoder pair (not shown).

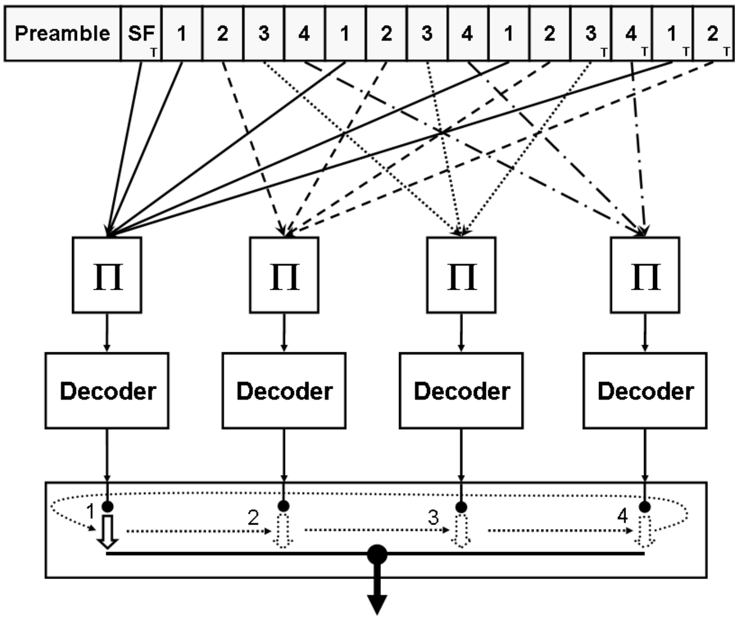


Figure 32: Exemplary TDMA-based stream arrangement for 4 parallel code streams

Each (133,171) convolutional code stream must be terminated with at least six zero tail bits, as explained in Appendix D. Stream termination is marked with a "T" in Figure 32. To obtain a convenient data granularity for implementation, 8 bits (N_{tail}=1 byte) have been chosen for the narrowband and 32 bits (N_{tail} = 4 bytes) for the wideband OFDM system. Note that the additional tail bits constitute data overhead, which increases payload. Pad bytes, which carry no information, are used to fill the last OFDM symbol. Let I_{dm} be the data mode index. The number of required code streams N_{stream} is a function of the message length N_{xrsc} (given in bytes), of the number of message bytes per OFDM data

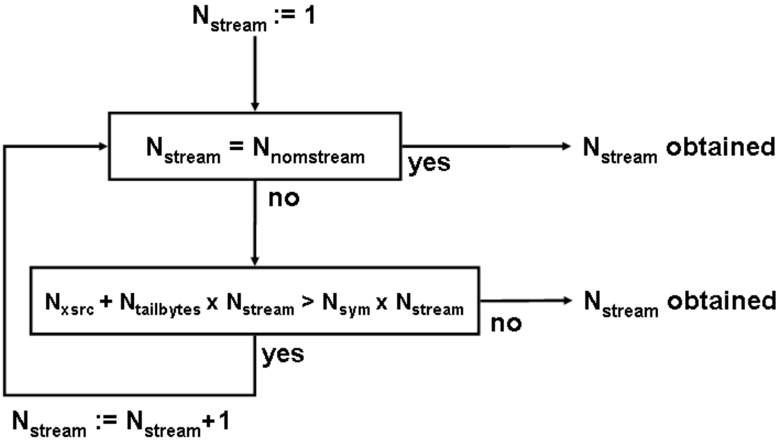


Figure 33: Determination of number of streams

symbol for mode I_{dm}, $N_{\mathrm{sym}}(I_{\mathrm{dm}})$, and of $N_{\mathrm{nomstream}}(I_{\mathrm{dm}})$.

$$N_{\mathrm{stream}} \equiv f\left(N_{\mathrm{xrsc}}, N_{\mathrm{sym}}(I_{\mathrm{dm}}), N_{\mathrm{nomstream}}(I_{\mathrm{dm}})\right) \tag{106}$$

$N_{\mathrm{nomstream}}(I_{\mathrm{dm}})$ denotes some stored number of code streams usually applied by the transmitter to at least fulfill the throughput requirement when using data mode I_{dm} for arbitrarily long message length [1]. The transmitter can determine $f(...)$ by applying a simple algorithm shown in Figure 33. This algorithm has been introduced to account for short message lengths, where not all of the $N_{\mathrm{nomstream}}(I_{\mathrm{dm}})$ code streams are getting used. It starts with the hypothesis that one stream is sufficient ($N_{\mathrm{stream}} = 1$) and increments N_{stream} until either the full stream number $N_{\mathrm{nomstream}}(I_{\mathrm{dm}})$ is reached or N_{stream} streams for $N_{\mathrm{stream}} < N_{\mathrm{nomstream}}(I_{\mathrm{dm}})$ are sufficient to encode the message using only one OFDM symbol per stream. For N_{tail} tail bytes, the number of required OFDM data symbols is then expressed via Equation (107). This number does not yet include optional midamble symbols.

$$N_{\mathrm{OFDM}} = \left\lceil \frac{N_{\mathrm{xrsc}} + N_{\mathrm{tail}} N_{\mathrm{stream}}}{N_{\mathrm{sym}}(I_{\mathrm{dm}})} \right\rceil \tag{107}$$

If optional outer Reed-Solomon encoding is applied, N_{xrsc} denotes the extended message length after this encoding. In other respects it has no influence on stream arrangement. The RS encoder requires a complete message block for encoding, which must be provided by another portion of pad bytes.

To avoid potential large overhead of up to 238 bytes for the RS (239,255) encoder, the policy can be to avoid encoding of the last block in adverse cases. N_{block}, N_{src}, N_{msg} and N_{res} denote the number of required blocks to encode all message bits, the message data length in bytes, the amount of message bytes per block and the amount of remaining message bytes in the last block respectively. Then, $N_{block} = \lceil N_{src}/N_{msg} \rceil$ and $N_{\mathrm{res}} = N_{\mathrm{src}}$ modulo N_{msg}. Then the policy regarding the last

---

[1] For example, 16-QAM with code rate 1/2 in the narrowband mode has a PHY data rate of 480 MBit/s and requires at least $N_{\mathrm{nomstream}} = 5$ code streams if decoders are running at 100 MHz

block may be defined as follows, if $N_{\mathrm{res}} > 0$: the last block is only encoded, if $N_{\mathrm{res}} \geq N_{\mathrm{min}}$ and $(N_{\mathrm{msg}} - N_{\mathrm{res}})/(N_{block} N_{\mathrm{msg}}) < \gamma$. The first condition prevents encoding for just a few residual bytes. The second condition comes into play for very short packets. It ensures that overhead does not exceed a certain fraction γ of the data frame. Otherwise this overhead may lead to higher loss than coding gain in terms of power efficiency and also greatly reduce the data rate. γ can be set to different values for different data modes to include the preamble loss in the consideration.

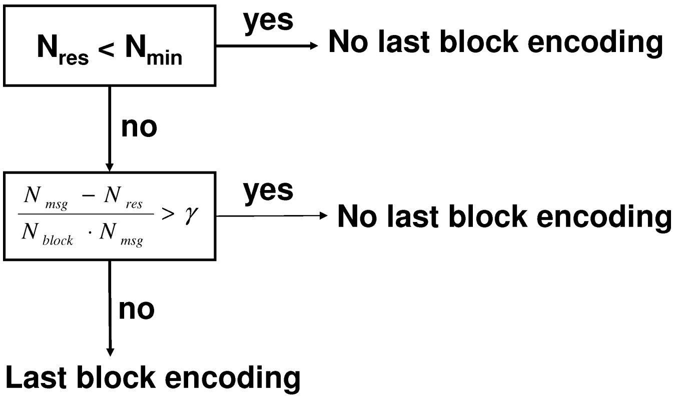


Figure 34: Determination for RS outer encoding of last block

The general frame structure of the wideband (EASY-A) PHY is depicted in Figure 35. The preamble consists of two parts and has a length of 6.6 OFDM regular OFDM symbols or 3.9 μs. It is followed by the signal field (SF) and the actual data frame. The signal field carries all parameters of the current frame. The encoded data of the SF is terminated in order to allow immediate convolutional decoding. RS encoding is not applied, because the following data cannot be processed before the SF is interpreted. To achieve higher robustness, every code bit of the SF is simply doubled (repetition code of length 2), since SF data constitutes less than 50% of available data space in the symbol.

Midamble parts with a length of two OFDM symbols can be optionally inserted into the data frame for channel re-estimation. The rate of the midambles is a variable parameter specified in the SF.

The frame structure of the narrowband PHY, which was developed first, has a few different parameters. The preamble length is extended to 11 OFDM symbols. Instead of midambles, low-constellation data symbols using either BPSK or QPSK modes can be inserted for decision-feedback channel re-estimation. This scheme is not further discussed here, since it could not be implemented due to the high latency of the feedback loop.

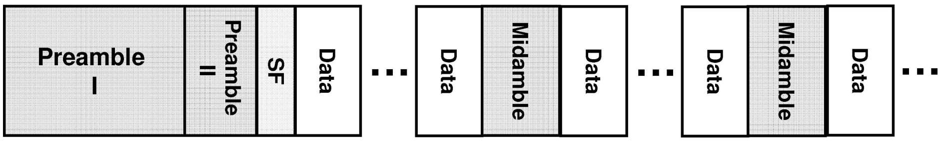


Figure 35: General EASY-A frame structure including preamble and midambles

5.5 Interleaver design and convolutional code performance

In the context of OFDM combined with convolutional codes, interleaving is mandatory for good system performance. Before discussing interleaving, we first make some conventions regarding indexing.

We adopt the bit indexing of the 802.11a standard to address bits within an OFDM symbol. N_{dsc} shall denote the number of data subcarriers and m_{sc} the fixed number of mapped (or loaded) code bits per subcarrier for the particular mode (BPSK, QPSK etc.). At first, we introduce a subcarrier index I_{sc}, $0 \leq I_{sc} < N_{dsc}$, so that the subcarrier with the most negative frequency has an index of $I_{sc} = 0$ and the highest an index of $I_{sc} = N_{dsc} - 1$. The content of an OFDM symbol is a bit vector of length $N_{\text{block}} = N_{dsc} \cdot m_{sc}$. Each bit is addressed with an index I_b, $0 \leq I_b < N_{\text{block}}$. The subcarrier of bit I_b is the one with the index $I_{sc} = \lfloor I_b / m_{sc} \rfloor$, i.e. the first m_{sc} bits belong to subcarrier $I_{sc} = 0$ and so forth. In addition, we define a constellation index I_c, $0 \leq I_c < m_{sc}$ to address the different bits of a subcarrier. This index is given as $I_c = I_b \bmod m_{sc}$ where "mod" denotes the conventional modulo operation. For all modes higher than BPSK, m_{sc} is even. The convention is to first count the bits mapped on the inphase-component (I_c = 0...$m_{sc}/2 - 1$) and then the bits in the quadrature component (I_c = $m_{sc}/2$...$m_{sc} - 1$). In both cases, the most reliable bit, or "MSB", is indexed first, then the next reliable bit down to the LSB[1].

Interleaving is required to even out the link quality variations affecting the transmitted bits. If many successive code bits suffer from low SNR, a wrong path decision of the Viterbi decoder is more likely to appear. Without interleaving, consecutive coded bits are mapped incrementally in frequency domain. For frequency selective channels, groups of adjacent code bits could therefore experience a low SNR.

5.5.1 Standard 802.11a-type interleaver

The first attempt for the narrowband PHY consisted in a reuse of the standard IEEE 802.11a interleaver. This interleaver consists of two permutations. The first performs tone interleaving in order to spread adjacent code bits across the spectrum and in this way avoid the aforementioned block fading problem. The second permutation is used to ensure that successive bits are mapped on bit positions with alternating reliability. Unequal bit reliabilities appear for 16-QAM and 64-QAM[1]. The bit vectors associated with the original content, the arrangement after the first and second permutation are denoted as $\vec{u}$, $\vec{v}$, $\vec{w}$ with elements $u(i)$, $v(j)$, $w(k)$, $0 \leq \{i, j, k\} < N_{\text{block}}$. The free parameter of this interleaver is given by the "hop width" Δ_{sc}, the subcarrier distance of adjacent code bits. With $N_{\text{cycle}} = N_{dsc}/\Delta_{sc}$, $\Delta_{bit} = m_{sc} \cdot \Delta_{sc}$ and $s = \max(m_{sc}/2, 1)$, the first and second permutation are defined as

---

[1] See constellation mappings in Appendix E and bit metrics in Figure 93, Appendix D.3.

$$v(j = f_{uv}(i)) := u(i) \tag{108}$$

$$w(k = f_{vw}(j)) := v(j)$$

$$f_{uv}(i) = \Delta_{bit} \cdot (i \bmod N_{\text{cycle}}) + \lfloor i/N_{\text{cycle}} \rfloor \tag{109}$$

$$f_{vw}(j) = s \cdot \lfloor j/s \rfloor + ((j + N_{\text{block}} - \lfloor j/\Delta_{bit} \rfloor) \bmod s) \tag{110}$$

Permutation rule (109) is explained as follows. Δ_{bit} is the bit distance of two subcarriers with frequency distance $I_{sc}^{(2)} - I_{sc}^{(1)} = \Delta_{sc}$. This means that any two bits with the same constellation index I_c have bit distance Δ_{bit}, when the first bit belongs to subcarrier $I_{sc}^{(1)}$ and the second to subcarrier $I_{sc}^{(2)}$. Subcarriers are essentially divided into "cycle groups" of N_{cycle} carriers, which lie on a grid with a width of Δ_{sc}. According to (109), the first N_{cycle} bits in $\vec{u}$ are mapped on the first constellation position $I_c = 0$ of each subcarrier in the first group, starting with subcarrier $I_{sc} = 0$. The next N_{cycle} bits are mapped on the second position $I_c = 1$ and so forth, until all positions in the first group are filled. The next $N_{\text{subblock}} = N_{\text{cycle}} \cdot m_{sc}$ bits in $\vec{u}$ are mapped onto the second group, which starts with subcarrier $I_{sc} = 1$, etc.

The second mapping (110) is a permutation of all those s bits, which belong to the inphase or quadrature component of their common subcarrier. This is necessary, since the first permutation gave every N_{cycle} adjacent bits the same reliability, where an alternating reliability is required in order to achieve a constant link quality. We think of the full block $\vec{v}$ being divided into contiguous sections of Δ_{bit} bits. We note that the first permutation mapped each of the N_{cycle} bits on some equivalent position of every next section. Therefore, it is sufficient to perform a common cyclical shift of all bits of the same section, if the cyclical shift step is linearly increased with the section index, and if this shift operation is done within the same axis (inphase or quadrature component). Indeed, the term $((j + N_{\text{block}} - \lfloor j/\Delta_{bit} \rfloor) \bmod s)$ leads to a circulation of the bits within the same axis, and the first term $s \cdot \lfloor j/s \rfloor$ positions each bit on the right subcarrier and axis (inphase or quadrature part).

It is instructive to consider a two-path channel having two strong received channel paths compared to all others. This case is not unlikely to happen. We remember that the direct path and first order reflections dominate at 60 GHz. The direct path and the ground reflection can provoke such a situation. Two paths with a relative delay of $\Delta\tau$ produce a comb filter with attenuated regions having frequency separation $\Delta f = 1/\Delta\tau$. Since the cycle groups have constant hop width, one could assume that a cycle group could be heavily attenuated for $\Delta f = \Delta f_{sc} \cdot \Delta_{sc}$. But the embedded pilot subcarriers with subcarrier distance 14 introduce an irregularity and prevent such a case[1]. Hence, this problem seems not to exist with the 802a interleaver as long as pilots are embedded.

5.5.2 Convolutional code performance using standard 802.11a-type interleaver

The performance of the standard convolutional code (171,133) has been investigated for the HHI channel model[1] under idealized conditions and was published in [Max06]. In that paper, the author com-

---

[1] But we also note that all pilot subcarriers could be attenuated, an aspect deserving further investigation.

[1] see Section 3.2

pared two different OFDM systems. Both systems feature a subcarrier spacing of 1.525 MHz and a guard time of 160 ns. The first system is the basic WIGWAM PHY. The second uses around twice the bandwidth (655 MHz compared to 333 MHz) by employing 384 subcarriers. Basically, this comparison was done to investigate the performance increase due to higher frequency diversity. All defined data modes were investigated, basic modes and such using puncturing in order to obtain intermediate rates. For puncturing, the standard patterns have been applied given in Appendix D.2. The conditions for the performed Monte-Carlo simulations are summarized below.

- The only modeled link effects had been the radio channel and white Gaussian noise. The system was simulated for the line-of-sight (LOS) case, since Si-Ge PA output power was regarded as too limited for NLOS transmission.

- No pilots were included in the simulation, and no phase correction had been performed. Also, no preamble was transmitted and the guard interval was not counted in the bit energy E_b. We can shift the curves by 1.3 dB in x-direction to account for these losses.

- Perfect time synchronization was established using Equation (66), (unnecessary) frequency synchronization was omitted.

- Zero-forcing channel equalization was done with perfect channel state information.

- Simplified soft-bit metrics (Appendix D.3) were used without quantization.

- The channel impulses were only jointly normalized as explained in Section 3.2. This is adequate for fixed-power transmission accounting for fading due to small-scale displacement.

- The frame size was chosen to 1024 bytes. 1000 frames were simulated for each SNR value.

- A hop width of $\Delta_{sc} = 12$ (18.75 MHz) for the narrowband and $\Delta_{sc} = 24$ (37.5 MHz) for the wideband system was found to perform best.

Simulation results for the optimized hop widths are shown in figures 36 and 37 for the two different antenna configurations. Several conclusions can be drawn.

- As described in section 3.2, a Vivaldi antenna with a gain of approximately 12 dBi is used for the receiver. This antenna accentuates the LOS path coming from the main direction, whereas first and second order reflection waves are likely to arrive with angles outside the main lobe. As a result, the reflection waves appear attenuated compared to the LOS path and the channel impulse response becomes more similar to a Dirak channel. In frequency domain this means that the channel is flattened and appears less frequency selective. Hence, system II cannot benefit that much from higher frequency diversity and both systems show quite similar performance.

- For the omni-omni link, more frequency selectivity and power fluctuations are present. System II, with its wider bandwidth, outperforms system I. The difference varies from 1.0 to 1.8 dB at a frame error rate of 10%. The high power fluctuations for system I were already noticed in Figure 10.

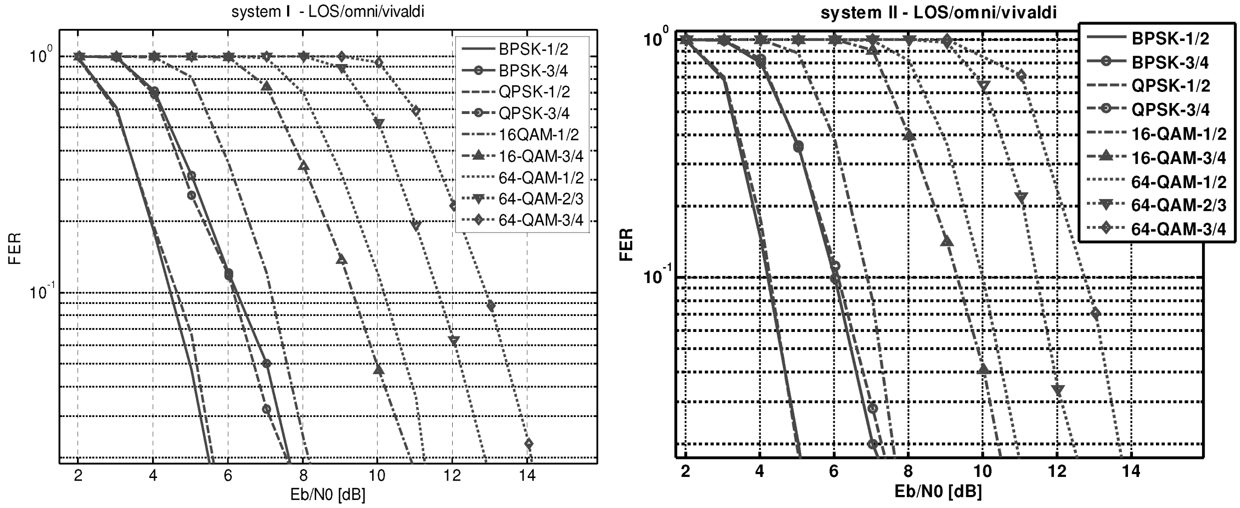


Figure 36: convolutional code performance with standard interleaver in omni-Vivaldi link scenario

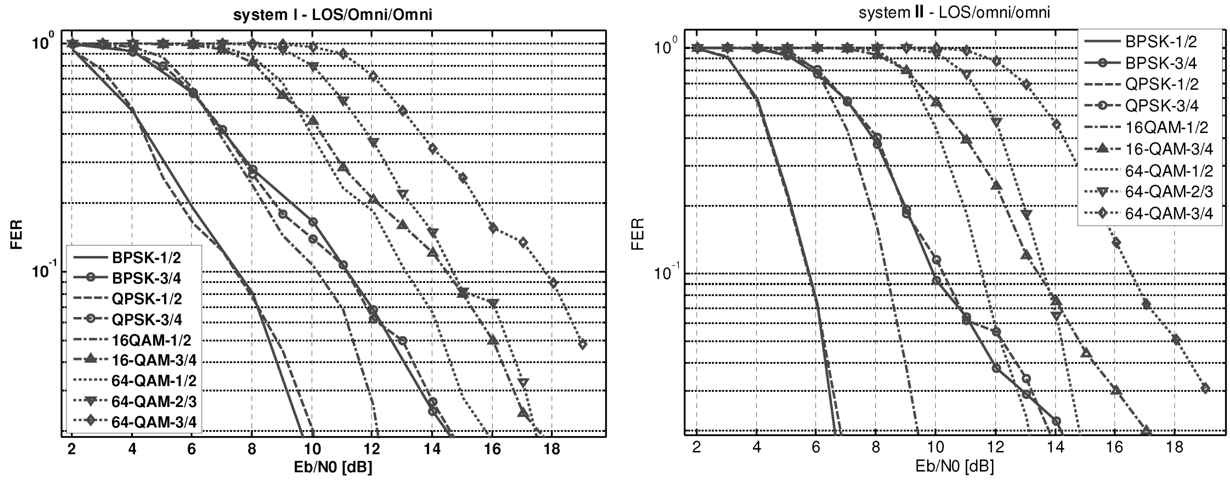


Figure 37: convolutional code performance with standard interleaver in omni-omni link scenario

- In general, punctured encoding reveals to have a bad performance for the omni-omni link. It seems that it cannot cope well with higher frequency selectivity of the channel. 64-QAM transmission with rate r=1/2 (no puncturing) offers the same data rate as 16-QAM, r=3/4, but shows much better performance for low FER. This 64-QAM mode was not specified in 802.11a but can be advantageously employed for the 60 GHz system.

We note that the HHI channel model was available prior to the TG3c model. With rms channel delay spreads of no more than 4 ns for the omni-Vivaldi-LOS channel (see Figure 4 in Section 4), the advantage to use OFDM is not so obvious as for the TG3c model. For the latter, the office model CM3.1 shows much more channel dispersion. Also, the advantage becomes more evident for a bandwidth of two GHz, meaning that a single-carrier system would have a five times shorter symbol time.

5.5.3 "Folded" interleaver

Choosing a very short hop width Δ_{sc} smaller than the coherence bandwidth of a channel with very short delay spread leads to adjacent code bits suffering from fading. Very large hop widths result in small cycle groups, since $N_{\text{cycle}} = N_{dsc}/\Delta_{sc}$. In this case, the code bit coming N_{cycle} positions later is mapped on the same subcarrier (for all modes higher than BPSK). Both cases rise the probability to experience burst errors. An optimum can be found to maximize code performance. We can add a third permutation to the standard interleaver, which enables longer groups without the penalty of a small hop width. We call this interleaver a folded interleaver. An additional free parameter, the folding factor n_{fold} provides a wider range of permutation schemes. At first, we write the combined permutation rule of the standard interleaver as a single function of the input index and hop width Δ_{sc}.

$$\begin{aligned} & w(k = f_{vw}(f_{uv}(i))) = v(f_{uv}(i)) = u(i) \\ & \Rightarrow k = f_{vw}(f_{uv}(i)) \equiv f_{uw}(i, \Delta_{sc}) \end{aligned} \tag{111}$$

The first permutation consists of the standard permutation for a fractional hop width $\Delta_{sc2} = \Delta_{sc}/n_{\text{fold}}$.

$$w(k) := u(i), \quad k = f_{uw}(i, \Delta_{sc2}) \tag{112}$$

We now reconsider the standard permutation which was defined in the beginning of the section. Recall the definitions made there. We can think of the bitvector after the standard interleaver being divided into $N_{\text{cycle2}} = N_{dsc}/\Delta_{sc2}$ regions of $\Delta_{bit2} = m_{sc} \cdot \Delta_{sc2}$ bits. The first permutation (109) distributes the elements of each cycle group into different regions, so that each element falls into one region. The second permutation (109) permutes bits only within their region.

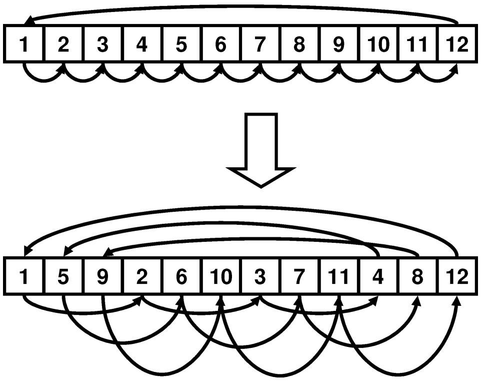


Figure 38: Third permutation performed over regions for $n_{\text{fold}} = 3$

Now we perform a third permutation of whole regions as shown in Figure 38 (for 12 regions and $n_{\text{fold}} = 3$). With $N_r = N_{\text{block}}/n_{\text{fold}}$ and $\Delta_{bit} = m_{sc} \cdot \Delta_{sc}$, this is achieved with the following permutation rule.

$$\begin{aligned} & y(l = f_{wy}(k)) = w(k) \\ & f_{wy}(k) = k \bmod \Delta_{bit2} + \lfloor (k \bmod N_r)/\Delta_{bit2}) \rfloor \cdot \Delta_{bit} + \lfloor k/N_r \rfloor \cdot \Delta_{bit2} \end{aligned} \tag{113}$$

This permutation restores the old hop width Δ_{sc} and produces cycle groups of length N_{cycle2} which are n_{fold}-times longer than for standard interleaving. This is achieved by folding of the cycle groups as shown in Figure 39, this time for $n_{\text{fold}} = 2$ and just a few subcarriers.

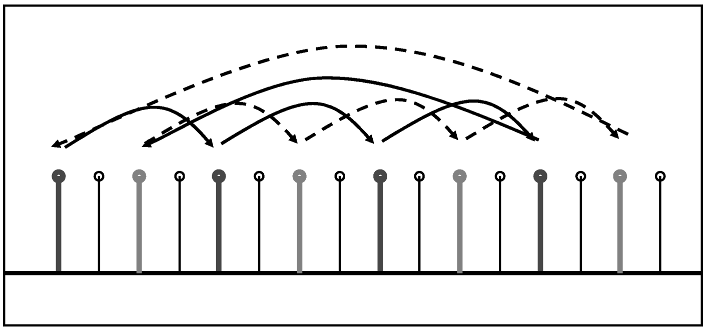


Figure 39: Folded cycle group $n_{\text{fold}} = 2$ ([Max09a])

5.5.4 Performance comparison for standard and folded interleaver

The author has compared the performance of the standard interleaver to the folded one for the residential NLOS channel CM2.3 and wideband OFDM parameters as given in Appendix G. Simulations were done for a complete receiver model including synchronization, channel estimation and tracking.

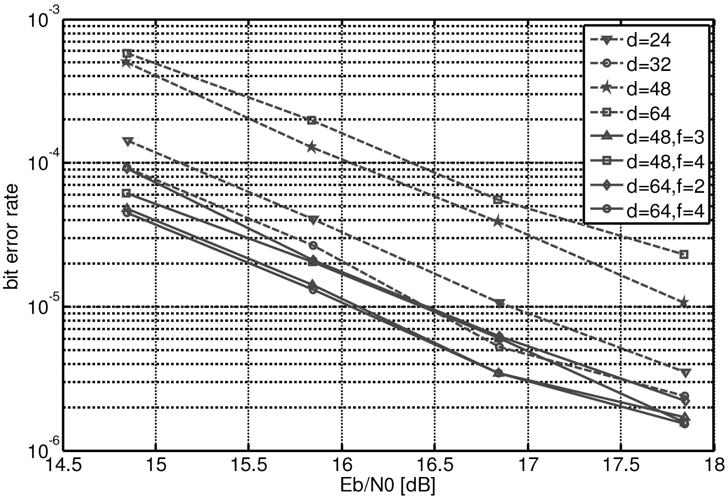


Figure 40: Interleaver comparison for CM2.3 and EASY-A PHY parameters ([Max09a])

The results are shown in Figure 40 and were presented in [Max09a], where d and f are the hop width and folding factor, respectively. The standard interleaver achieves its performance maximum for $\Delta_{sc} = 32$ (or $\Delta f \cdot \Delta_{sc} = 2.11 \text{ MHz} \cdot 32 = 67.5 \text{ MHz}$). Two of the folded interleavers show better performance. The difference is about 0.5 dB at a BER of 10^{-5}. This performance increase seems to be little, but comes at no additional expense when using a memory-based, general-purpose (de-)interleaver architecture as presented in Section 6.9. For the final system, the folded interleaver with the parameters $\Delta_{sc} = 64$ and $n_{\text{fold}} = 4$ has been selected.

5.5.5 Interleaving scheme for WiMAX LDPC code (768,384), performance comparison

For system comparison, an LPDC block code has also been also investigated for the EASY-A PHY, since LPDC codes can provide high coding gain at moderate complexity ([R. 62]) and became popular during the last decade. The selected code has been the (768,384) code found in the WiMAX standard ([WiM05]), which has a fixed code rate of r=1/2. Furthermore, an ASIC implementation was done with IHP inhouse technology, and results were published in [M. 10]. The decoder uses the standard min-sum algorithm ([D. 04]).

As such, an LDPC code may not require interleaving when combined with OFDM. On the other hand, the decoder is designed for a fixed code block length which is equal to the number of subcarriers in an OFDM symbol. With this arrangement, the number of code blocks per OFDM symbol is equal to m_{sc}, the number of mapped code bits per subcarrier.

When the code blocks would appear one after each other in the input stream, they would be mapped on separate frequency sections without interleaving and could, therefore, not benefit from full frequency diversity. Therefore, we introduce an interleaver consisting of two permutations. The first permutation (115) maps one code bit of each code block on every subcarrier, i.e. each subcarrier contains a bit from every code block after this stage. The second, (116), permutes bits within their subcarriers to equally distribute higher and lower bit reliability among the code blocks. Following the same notation as in (108), permutations are written as follows.

$$v(j = f_{uv}(i)) := u(i) \tag{114}$$

$$w(k = f_{vw}(j)) := v(j)$$

$$f_{uv}(i) = (i \bmod N_{dsc}) \cdot m_{sc} + \lfloor i/N_{dsc} \rfloor\,, \tag{115}$$

$$f_{vw}(j) = \lfloor j/m_{sc} \rfloor \cdot m_{sc} + ((j - \lfloor j/m_{sc} \rfloor) \bmod m_{sc}) \tag{116}$$

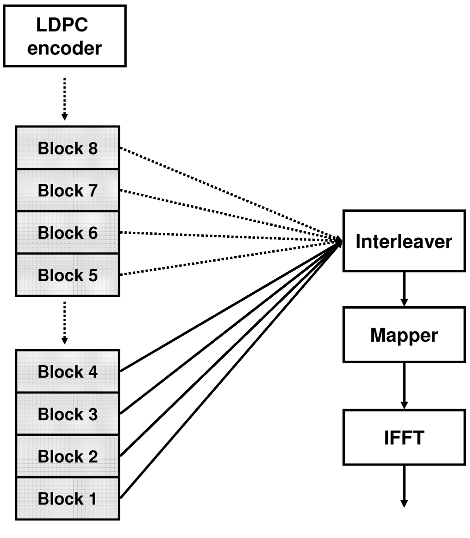


Figure 41: Transmitter signal flow for LDPC encoded blocks for 16-QAM

It is of interest to compare the performance of coded OFDM using the concatenated Reed-Solomon and convolutional code scheme versus this LDPC code. The first results obtained by the author were published in [Max09a] for 16-QAM and the TG3c CM 2.3 NLOS channel model. Simulations were done for the complete receiver model featuring all algorithms. On the other hand, no quantization effects were considered. Due to the RS encoder, the code rate of the convolutional scheme is slightly below the LDPC code rate. Results show that the RS-convolutional scheme slightly outperforms the chosen LDPC code, but the difference is in the range of 0.5 dB or below. Hence, the performance can be regarded as very similar and other factors like decoder silicon area need to be included in the system design decisions.

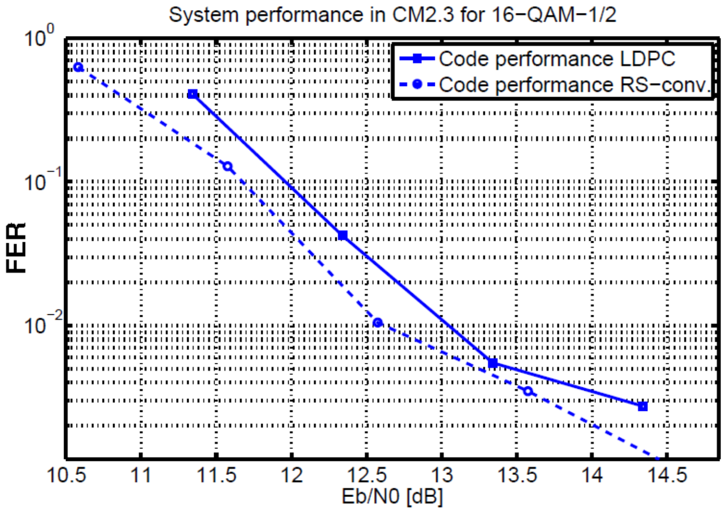


Figure 42: LDPC versus RS-convolutional code scheme for 16-QAM r=1/2 in CM2.3 ([Max09a])

5.6 Preamble waveform design

Before dealing with synchronization, we consider first the creation of preamble sequences. Following the same approach as in the WLAN standard 802.11a, the general preamble waveform is composed of a number of shorter sample sequences A and longer sequences B. Sequences A are used for AGC, coarse frame detection and frequency offset correction, while sequences B are intended for channel estimation, fine adjustment of window timing and may also be used for improved CFO correction. The question arose how these sequences can be created and what kind of optimization should be applied. Three different criterions have been regarded of particular interest. Other criterions were not considered.

1. In time domain, each sequence should have a low peak-to-average-power ratio (PAPR) to prevent nonlinear distortion, which otherwise may reduce the estimation accuracy. A low PAPR may also enable the transmitter to send the preamble with higher power than the data frame.

2. Timing synchronization algorithms, which are based on cross-correlation, require a sequence with good autocorrelation properties.

3. In frequency domain, the sequence energy should be evenly distributed. From a statistical point of view, a even spectrum leads to higher robustness against frequency selective fading.

To fulfill the third criterion, the prototype sequences A or B can be created in frequency domain. The corresponding time domain sequence is then obtained by inverse FFT operation. In this way, sequences were obtained for IEEE 802.11a.

At first, we consider the short sequence A and assume that A has a length L_A, which divides the regular FFT size N_{FFT}, i.e. $N_{\mathrm{FFT}} = N_A \cdot L_A$. If A is repeated N_A times, we obtain a sequence which can be *technically* realized with the same IFFT operation as for ordinary data symbols. The condition for the shorter period of N_{FFT}/N_A is fulfilled if subcarriers only with index $k \cdot N_A$ are nonzero, for integer k. On the other hand, this restricts the number of repeated A-sequences in the preamble to be a multiple of N_A. To overcome this, parts of the repeated A-sequence can be periodically extended in time domain. Alternatively, the full preamble can simply be stored as a waveform in memory.

We define a subcarrier index mapping $I_A(n)$, $-N_{\mathrm{FFT}}/2 \leq I_A(n) < -N_{\mathrm{FFT}}/2$ for subcarrier symbol vector $\vec{S}_A$ with elements $S_A(n)$, $1 \leq n \leq N_{S_A}$. This means that each symbol $S_A(n)$ is mapped on subcarrier $I_A(n)$. As in [IEEa], the symbols can be chosen from a finite alphabet. If for example BPSK or QPSK is chosen, then either $S_A \in (-1, 1)$ or $S_A \in (\pm 1 \pm j)/\sqrt{2}$.

For the long sequence B, we can apply the same approach. The difference consists in the number of subcarriers N_{S_B} and their index mapping $I_B(n)$. Since sequence B is intended for channel estimation, the subcarrier symbols $S_B(n)$ are mapped onto the subcarriers reserved for regular data *and* pilot symbols (of the continuous pilots).

To arrive at a low PAPR, a *sequence search algorithm* has been elaborated, which tries to find a minimum sequence PAPR. Before presenting this algorithm, it shall be mentioned that the algorithm normally finds lower PAPR sequences for BPSK than for QPSK, although QPSK gives two more choices for each subcarrier symbol. This happens due to the symmetry property of the DFT. We denote the time sequence as $\vec{s}$ with elements $s(m)$, $-N_{\mathrm{FFT}}/2 \leq m \leq N_{\mathrm{FFT}}/2 - 1$. If BPSK is chosen, the sequence $\vec{S}$ in frequency domain constitutes a *real*, not complex vector. Then the time sequence has the property that $s(m) = \overline{s(-m)}$. It also follows that the amplitude is symmetric with respect to the origin, $|s(m)| = |s(-m)|$. The algorithms yields better results for shorter sequences and therefore benefits from the symmetry. For this reason, QPSK has been dropped and only BPSK sequences are considered. The simple try-and-error algorithm is sketched below. The symbol alphabet is denoted as M_A.

1. Start with an arbitrary symbol vector $\vec{S}$, $S(n) \in M_A$.

2. Map the symbols $S(n)$ on subcarriers $I(n)$ and perform IFFT operation to obtain the time sequence $s(k)$. Calculate the PAPR of $s(k)$.

3. There are N_q global iterations with index q = 1...N_q. For every index q, perform $\mathrm{IT}(q)$ iterations of the following inner loop, using a step size of $\mathrm{ST}(q)$:

(a) Create a second sequence $\vec{S}_2 := \vec{S}$ with the same elements.

(b) Choose $\mathrm{ST}(q)$ random positions m_l, $1 \leq l \leq \mathrm{ST}(q)$ and assign a new random value $v_l \in M_A$ for the m-th symbol of $\vec{S}_2$, $S_2(m_l) := v_l$.

(c) Map the symbols $S_2(n)$ on subcarriers $I(n)$ and perform IFFT operation to obtain the time sequence $s_2(k)$. Calculate the PAPR of sequence $s_2(k)$.

(d) If the PAPR of sequence $\vec{s_2}$ is lower than the PAPR of the old sequence $\vec{s}$, update (replace) sequence $\vec{s}$ with $\vec{s}_2$ and $\vec{S}$ with $\vec{S}_2$ respectively.

Typical parameters have been chosen to $N_q = 5$, $IT(q) = 15000$ for all q and $ST = (5, 4, 3, 2, 1)$. Hence, the step size is monotonically decreased for every global iteration. In addition, the optimization routine can be manually restarted for the last sequence used as the initial sequence for the new run. This is done until no further improvement is achieved. Since the algorithm only finds a *local minimum* for the PAPR, the algorithm needs to be restarted several times for new initial vectors $\vec{S}$.

An accurate estimation of the PAPR requires *oversampling* of the sequence waveform. Oversampling has been realized in the algorithm as follows. $(N_{\mathrm{ov}} - 1) \cdot N_{\mathrm{FFT}}$ zero subcarriers are symmetrically stuffed at indices $|I| > N_{\mathrm{FFT}}/2$ to create an extended frequency vector of size $K_{\mathrm{ov}} = N_{\mathrm{ov}} \cdot N_{\mathrm{FFT}}$. This vector is transformed to frequency domain using a K_{ov}-point IFFT.

The obtained PAPR values for wideband and narrowband OFDM modes are shown in table 7. 16-times oversampling had been applied for high accuracy. The amplitude plot of a typical OFDM frame for the wideband mode consisting of the preamble (22 A-sequences and two B-sequences plus guard time) and 14 data symbols is shown in Figure 43. Preamble and data part have equal average power of P=1 (0 dBu).

The next four figures display the normalized cross-correlation of sequences A and B for the wideband mode. On the left side some section is zoomed, where each sequence appears in the preamble, whereas on the right the correlation over the whole preamble is plotted. It can be seen that the correlation properties for this class of sequences are good in both cases, but the cross-correlation does not immediately fall off to small values but has a gradual decay around the peak value.

Table 7: PAPR of short and long preamble sequences for narrowband and wideband OFDM mode

Type	Length	number of used subcarriers	PAPR
Short sequence (NB)	64	52	3.2 dB
Long sequence (NB)	256	208	3.8 dB
Short sequence (WB)	256	208	3.9 dB
Long sequence (WB)	1024	828	5.2 dB

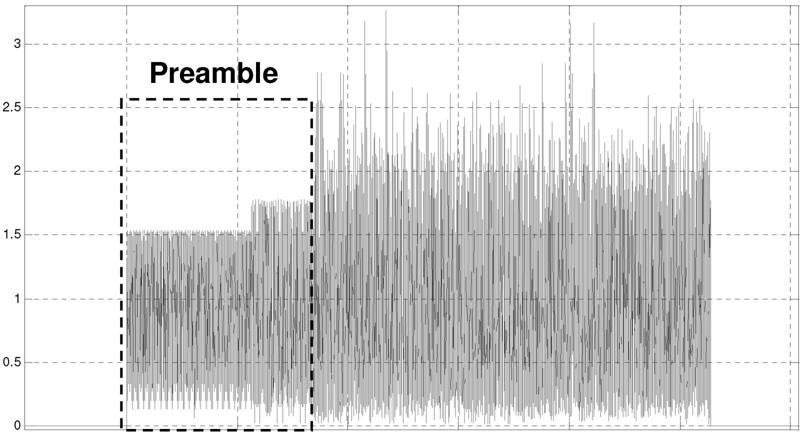
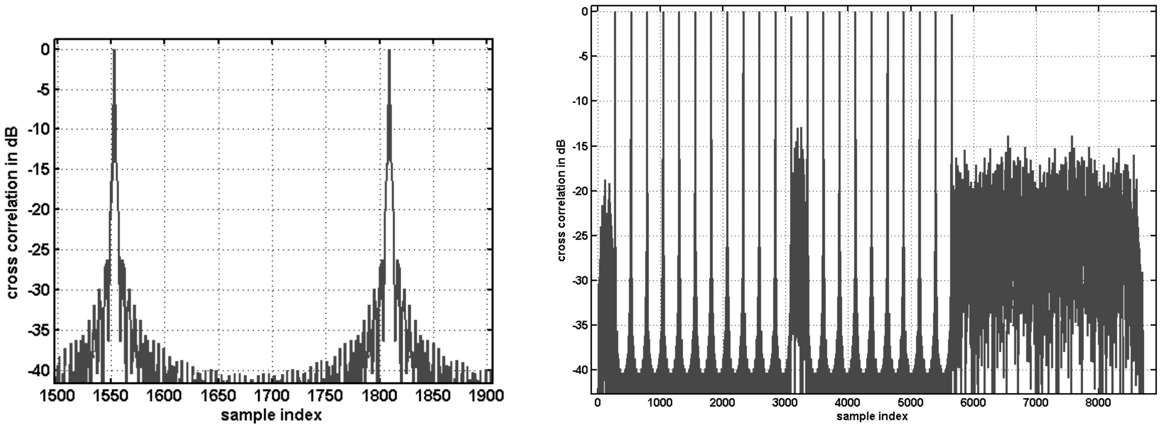


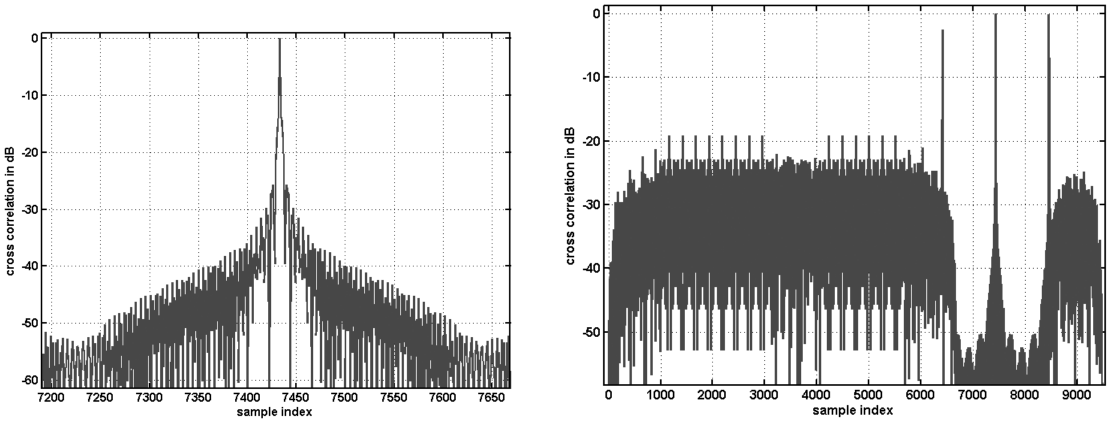
Figure 43: Amplitude waveform of typical PHY frame for wideband OFDM mode



(a)

(b)

Figure 44: Cross-correlation with short sequence A for wideband mode



(a)

(b)

Figure 45: Cross-correlation with long sequence B for wideband mode

5.7 Preamble processing overview

The author has published the preamble processing algorithm in a first version in [M. 07], but this work presents an improved and more comprehensive version. Note that the term "DFT" and "FFT" for this and the following sections do mean the same mathematical function, even though the FFT uses an optimized algorithm for computation (Appendix F). Therefore, these terms are interchanged when only the mathematical function is concerned.

Standard solution

For the beginning, we recall the standard solution for the IEEE 802.11a preamble ([IEEa]). In that scheme, autocorrelation is performed on the first preamble part for frequency offset estimation and frame detection, and cross-correlation on the second preamble part to achieve a refined frame timing. Afterwards, channel estimation is carried out in the frequency domain utilizing the two symbols of the second preamble.

Each of the tasks depend on the previously performed ones. Frame detection or coarse frame timing is required to attain the instant, when the autocorrelator is evaluated for CFO estimation. In turn, the cross correlator will work properly only after CFO correction. Finally, the channel estimator will deliver channel coefficients based on refined frame timing after cross-correlation. We note that the cyclic prefix of the second preamble is extended to half the DFT size in order to greatly reduce ISI.

An algorithm of this type is found in [Alf04]. There, a plateau detector is used for frame detection. As the channel delay spread increases and SNR decreases, the initial coarse timing, the moment when frame detection is obtained, experiences a higher variance, which is not harmful as long as cross-correlation is additionally employed. Cross-correlation usually requires a larger gate count in hardware, even after down-quantization of the input waveform or correlation pattern.

New algorithm

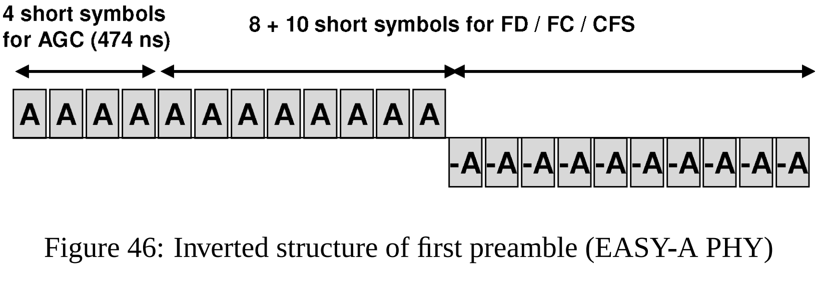


Figure 46: Inverted structure of first preamble (EASY-A PHY)

The developed synchronizer in this work is based on the idea to avoid cross-correlation and use fine timing estimation in frequency domain instead. Hence, the initial coarse timing needs to be precise enough to avoid timing errors for the start position of the DFT window. This is accomplished with a modified preamble structure, shown in Figure 46. It is also facilitated in the usual way with an extended guard time for the second preamble. The synchronization algorithm has a very low false-alarm probability enabling the usage of low thresholds.

An overview of all preamble processing steps is given in Figure 47. After frame detection and CFO correction, channel estimation and fine frame synchronization are performed in an interleaved way splitting channel estimation into two procedures. The first step consists of simple averaging of the preamble symbols for coarse channel estimation. The phases of the channels coefficients are used for fine frame timing. After changing the time reference, the channel estimation is refined with a frequency filter-based noise reduction method. The algorithms are treated in the following sections.

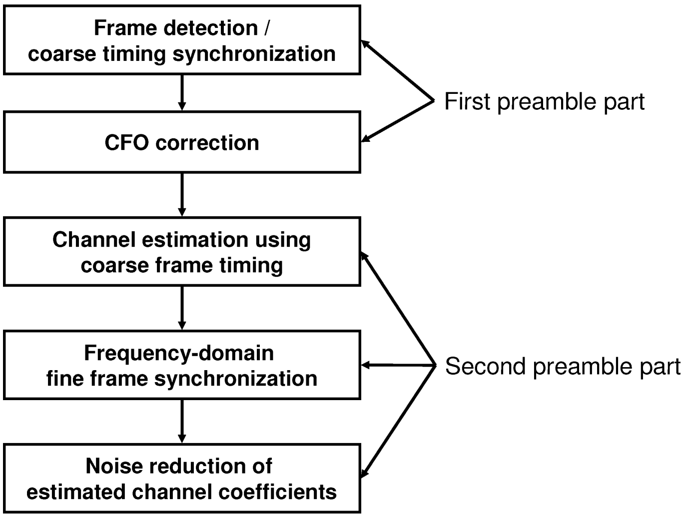


Figure 47: Preamble processing overview for wideband mode

5.8 Frame detection, coarse timing synchronization and CFO estimation

For frame detection, two exceptional cases can appear during operation. The first case is a miss of a new transmitted frame, if the signal is buried in noise. This can be circumvented to some extend with low threshold values for the detector. On the other hand, very low thresholds may provoke a *false alarm*. In this case, noise is erroneously detected as a frame. The presented algorithm achieves a very low false alarm probability for low thresholds.

Before discussing algorithms, we first focus on the autocorrelation function. Schmidl ([T. 97]) considered a timing metric $M(d)$ for the received signal r_n as defined in (117). The correlation sum is normalized to the power of the second symbol. A frame is detected if $M(d)$ exceeds a given threshold for at least a sample (or a plateau). The aim of the normalization in (117) is to obtain an invariant metric with respect to the signal level, and which produces small values $M(d)$ during AGC settling.

$$M(d) = \frac{|P(d)|^2}{(R(d))^2} \tag{117}$$

$$P(d) = \sum_{m=0}^{L-1} (\overline{r}_{d+m} r_{d+m+L}) \qquad R(d) = \sum_{m=0}^{L-1} |r_{d+m+L}|^2$$

In this work, a different normalization has been used. An obvious choice would be to define a normalized "autocorrelation function" for a complex input signal $x(n)$ given in Equation (118), where

$P_x(n, N_I)$ is the energy in an interval of duration N_I ending at sample index n (causal formulation).

$$\widehat{A}_x(n, N_\Delta, N_I) = \frac{\sum_{k=0}^{N_I-1} \overline{x(n-k-N_\Delta)}x(n-k)}{\sqrt{P_x(n-N_\Delta, N_I)P_x(n, N_I)}} \tag{118}$$

$$P_x(n, N_I) = \sum_{k=0}^{N_I-1} \overline{x(n-k)}x(n-k) \tag{119}$$

Using Cauchy-Schwarz inequality, we have $|\widehat{A}_x(n, N_\Delta, N_I)| \leq 1$, and $|\widehat{A}_x(n, N_\Delta, N_I)| = 1$ is true if $x(n-k) = \alpha \cdot x(n-k-N_\Delta)$ for all $k = 0..(N_I - 1)$. Evaluating a threshold condition like $|\widehat{A}_x(n, N_\Delta, N_I)|^2 \geq \gamma$ avoids computation of square-root operation, but requires a high dynamic range for the implementation. For this reason, another metric has been used, where the autocorrelation is normalized to the mutual maximum of $P_x(n, N_I)$ and $P_x(n - N_\Delta, N_I)$.

$$A_x(n, N_\Delta, N_I) = \frac{\sum_{k=0}^{N_I-1} \overline{x(n-k-N_\Delta)}x(n-k)}{\max\{P_x(n, N_I), P_x(n-N_\Delta, N_I)\}} \tag{120}$$

From definition (120), it follows that $|A_x(n, N_\Delta, N_I)| \leq |\widehat{A}_x(n, N_\Delta, N_I)| \leq 1$ and $|A_x(n, N_\Delta, N_I)| = 1$ if and only if $x(n-k) = e^{j\phi_0} \cdot x(n-k-N_\Delta)$ for all $k = 0..(N_I - 1)$. Metric (120) satisfies the needs for the synchronization algorithm.

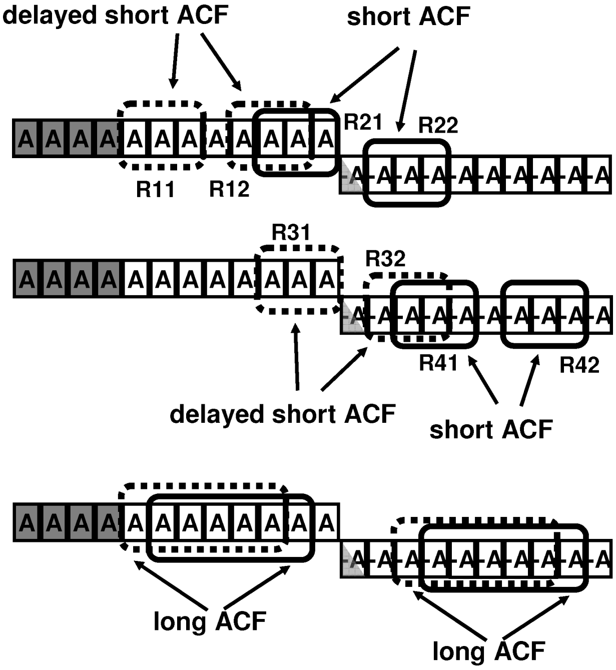


Figure 48: Short ACF and long ACF operating on first preamble

The received signal is denoted as $z_{RX}(n)$. We assume the link model shown in Figure 24 in Section 4.4. The original preamble is made up of non-inverted and inverted A-sequences. To keep notations simple, we denote these transformed sequences also as "A". The principle of operation is shown in Figure 48 for the wideband mode parameters. Two autocorrelators are used in parallel, a short ACF for frame detection and a long ACF for frequency correction. The algorithm is described with ACF parameters of the wideband mode. Generalizations using different parameters are straight forward. N_s denotes the length of the short sequence. Short ACF $\Psi_S(n)$ and long ACF $\Psi_L(n)$ shall be defined as

$$\begin{aligned}\Psi_S(n) &= A_{z_{RX}}(n, N_\Delta = 4N_s, N_I = 3N_s) \\ \Psi_L(n) &= A_{z_{RX}}(n, N_\Delta = N_s, N_I = 6N_s)\end{aligned} \tag{121}$$

After channel convolution, the first preamble retains its periodicity of N_s samples for a certain period of at least 8 A-sequences and another 9 inverted A-sequences. This is the case if the AGC has quickly settled and if the channel response $h(n)$ is shorter than the duration of an A-sequence, which is equal to the cyclic prefix. This means that only the first inverted A-sequence will be corrupted by the channel. Carrier-frequency offset $\triangle f_0$ causes an additional phase rotation over time. For the notation, we use $\triangle\Omega_0 = 2\pi f_0/f_T$, the normalized frequency deviation for sampling frequency f_T.

The short ACF is used to detect the frame exploiting the sign-flip of the A-sequences. n_{12}, n_{22}, n_{32}, n_{42} shall define the very last samples within the regions R_{12}, R_{22}, R_{32}, R_{42} in Figure 48. For very high SNR and negligible phase noise, the following approximations apply (regions R_{22} and R_{32} are identical).

$$\begin{aligned}z_{RX}(n) \approx&\ \ z_{RX}(n-4N_s)\cdot e^{j\triangle\Omega_0\cdot 4N_s} \quad \text{for} \quad n \in R_{12} \\ \Rightarrow \Psi_S(n_{12}) \approx&\ e^{j\triangle\Omega_0\cdot 4N_s} \\ z_{RX}(n) \approx&\ -z_{RX}(n-4N_s)\cdot e^{j\triangle\Omega_0\cdot 4N_s} \quad \text{for} \quad n \in R_{22} \\ \Rightarrow \Psi_S(n_{22}) \approx&\ -e^{j\triangle\Omega_0\cdot 4N_s} \\ z_{RX}(n) \approx&\ -z_{RX}(n-4N_s)\cdot e^{j\triangle\Omega_0\cdot 4N_s} \quad \text{for} \quad n \in R_{32} \\ \Rightarrow \Psi_S(n_{32}) \approx&\ -e^{j\triangle\Omega_0\cdot 4N_s} \\ z_{RX}(n) \approx&\ \ z_{RX}(n-4N_s)\cdot e^{j\triangle\Omega_0\cdot 4N_s} \quad \text{for} \quad n \in R_{42} \\ \Rightarrow \Psi_S(n_{42}) \approx&\ e^{j\triangle\Omega_0\cdot 4N_s}\end{aligned} \tag{122}$$

Therefore, the following relationships are approximately satisfied.

$$\begin{aligned}&\Rightarrow \Psi_S(n_{22}) \approx -\Psi_S(n_{12}) = -\Psi_S(n_{22} - 5N_s) \\ &\Rightarrow \Psi_S(n_{42}) \approx -\Psi_S(n_{32}) = -\Psi_S(n_{42} - 5N_s)\end{aligned} \tag{123}$$

These relationships are *independent of the frequency offset* and can be exploited for frame detection. This is done by relating the output of the ACF to the same signal, but delayed by $5N_s$ samples. Then we will observe two regions where

$$\Psi_S(n) \approx -\Psi_S(n - 5N_s) \tag{124}$$

and we will call these regions the "antiphase-peaks" or simply "peaks". Due to the noise contributions, the ACF signal will not have exactly the opposite phase with respect to the delayed signal as stated above and will be also decreased in amplitude. The synchronizer works in that way as to identify all time instants, where the short ACF signal satisfies the anti-phase condition (124) at least to some extend. The ACF signal is related to the delayed signal as follows:

$$\tilde{\Psi}_S(n) := -\Psi_S(n) \cdot \exp(-j\angle\{\Psi_S(n-5N_s)\}) \tag{125}$$

Here, $\angle\{z\}$ denotes the angle of the complex number z in radians. With (125), the angle of $\Psi_S(n)$ is related to the angle of $\Psi_S(n-5N_s)$ and flipped in sign. For the anti-phase case, $\tilde{\Psi}_S(n)$ will align with the real axis in the complex plane in positive direction. The following conditions are evaluated, which can be easily implemented with the aid of a CORDIC operation.

$$\begin{aligned} |\Psi_S(n-5N_s)| &> \alpha_1 \qquad (126)\\ \mathrm{Re}\{\tilde{\Psi}_S(n)\} &> \alpha_2 \qquad (127)\\ \alpha_3 \cdot \mathrm{Re}\{\tilde{\Psi}_S(n)\} &> |\mathrm{Im}\{\tilde{\Psi}_S(n)\}| \qquad (128)\\ &\Rightarrow n \in \mathcal{M} \end{aligned}$$

α_1, α_2, α_3 are appropriate positive threshold parameters. Condition (126) ensures a high amplitude for the delayed ACF and conditions (127) and (128) ensure that the rotated ACF $\tilde{\Psi}_S(n)$ has a peak in the direction of the positive real axis at a small deviation angle, given that α_3 is small. The samples, which satisfy these conditions, shall belong to a set $\mathcal{M}$.

Clustering

The next step consists in the separation of the acquired indices $n \in \mathcal{M}$ into different cluster sets $\mathcal{M}_i$. If these indices are grouped in ascending order $m_k \in \mathcal{M}, m_1 < m_2 < m_3 < ...$, then the assignment is done such, that adjacent indices, which belong to the same cluster $\mathcal{M}_i$ will differ in no more than d_1 positions:

$$(m_k \in \mathcal{M}_i) \wedge (m_{k+1} \in \mathcal{M}_i) \Leftrightarrow m_{k+1} - m_k \leq d_1 \tag{129}$$

Under normal circumstances, two index clusters will arise covering a region around n_{22} and n_{42}, where the anti-phase condition is met. With parameter d_1 set to a few samples, it is avoided that samples within the same region are wrongly partitioned into two or more clusters in case of very low SNR. This would only happen if a number $K > d_1$ of adjacent samples inside the region failed conditions (126)-(128) due to noise. The width of the clusters depends on the chosen threshold parameters and the length of the channel response. Clustering can be implemented with simple counters and some additional logic.

Coarse Frame detection and frequency estimation

We define a peak point p_i of cluster $\mathcal{M}_i$ as the truncated mean of the lowest and highest index within $\mathcal{M}_i$.

$$p_i = \lfloor (\min_{n \in \mathcal{M}_i}\{n\} + \max_{n \in \mathcal{M}_i}\{n\})/2 \rfloor \tag{130}$$

Two peaks must be found within the frame and these peaks must have a distance within some tolerance range. A frame is detected, if the following condition holds

$$d_2 - \Delta d \leq p_{i+1} - p_i \leq d_2 + \Delta d \tag{131}$$

p_i is taken as the time reference. The long autocorrelation, which is used for carrier frequency estimation, is then evaluated in a save region at $p_i - d_3$ before the sign flip and at $p_i + d_4$ after the sign flip where the A-sequence does not change for the full length of the long ACF (see Figure 48). The estimated CFO in radians per sample is taken from the averaged long ACF at the defined positions:

$$\Psi_{L_i} = [\Psi_L(p_i - d_3) + \Psi_L(p_i + d_4)]/2 \tag{132}$$

$$\triangle\hat{\Omega}_{0,i} = \frac{\angle\Psi_{L_i}}{N_s} \tag{133}$$

The frame is finally accepted, if the long ACF also exceeds a threshold α_4.

$$|\Psi_{L_i}| > \alpha_4 \tag{134}$$

For the considered wideband case, the critical threshold parameters have been optimized to the values given in table 8. α_3 is chosen quite large to avoid frame detection failure due to high phase noise. After the frame has been detected and the carrier frequency offset has been corrected using a numerical controlled oscillator (NCO), the two long symbols of the second preamble are evaluated for channel estimation and fine synchronization. The start position (see Figure 54 in Section 5.10) for the first DFT is chosen to

$$n_{\text{start}} = p_i + d_5 \tag{135}$$

d_5 is a parameter optimized to attain the lowest ISI for the considered channel models at high and very low SNR. At low SNR, the variance of the start position will increase. This is accounted with a earlier start position avoiding ISI at the end of the second symbol, but also reducing the effective guard duration.

Table 8: Detection parameters

parameter	value
α_1	0.4
α_2	0.4
α_3	1
α_4	0.5
d_1	48
Δd	128

Figure 49 shows the signal curves for the magnitude of the short and long autocorrelation for a Dirac-channel and a typical HHI Omni-omni NLOS channel without any noise. In both cases, two clear anti-phase peaks can be observed.

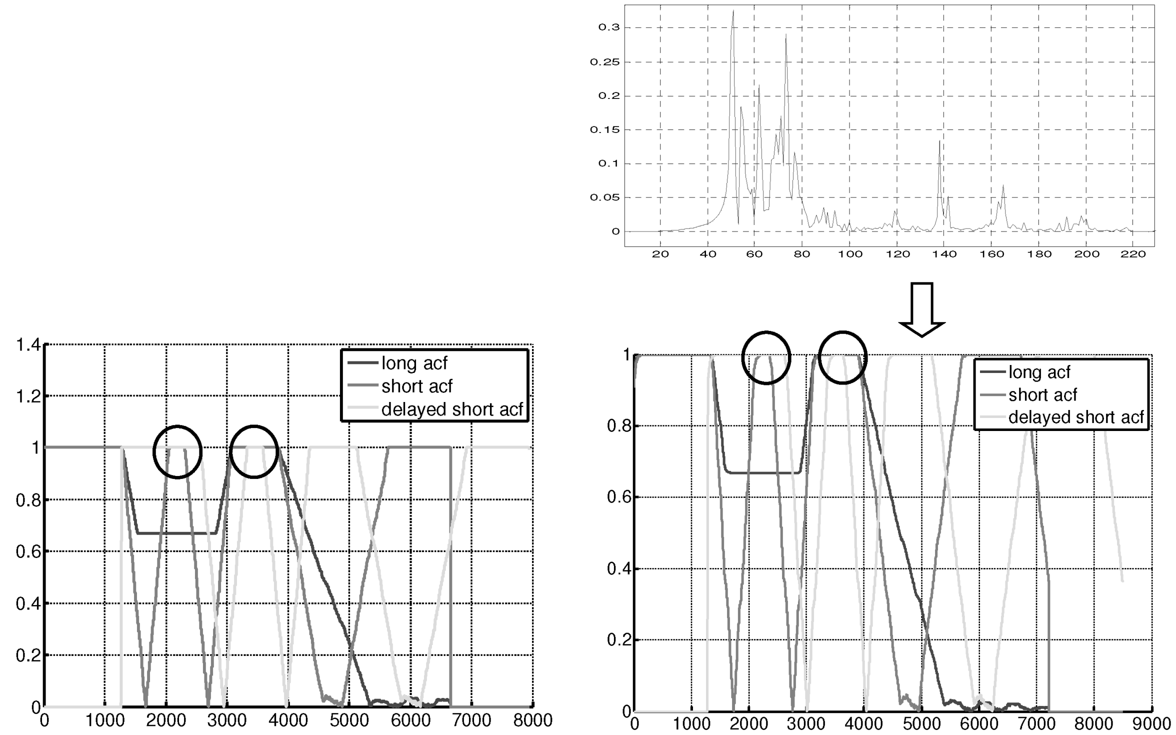


Figure 49: Short and long autocorrelation signals, anti-phase peaks

5.9 Channel estimation and equalization

The basics of OFDM transmission through linear (time-invariant) channels have been recalled in Appendix C. Given that the guard interval covers the full channel spread, each subcarrier at frequency $f = l\Delta f$ is multiplied with $H(f = l\Delta f)$, the Fourier transform of the channel impulse response $h(\tau)$ at this frequency.

5.9.1 Channel estimation

A common way to perform channel estimation is to transmit and utilize one to a few reference OFDM symbols, which contain training data. This data is transmitted as known symbols mapped on all active subcarriers (omitting the guard subcarriers). Channel coefficients are analyzed in frequency domain, and estimation gain can be achieved with averaging.

The 802.11a standard provides two identical reference symbols for channel estimation, similar to the lower preamble structure in Figure 50. This scheme using *identical* symbols has two advantages.

- Since periodicity is maintained over the whole preamble part, the second symbol does not need a separate cyclic prefix.
- Due to the linearity of the DFT, averaging can be carried out in time domain, and only one FFT needs to be performed, saving power.

The 802.11a preamble is protected by an extended guard time being twice as large compared to system guard time. This means that the channel estimation suffers from (far) less ISI, as long as proper time synchronization is established. The whole scheme has been adopted for both 60 GHz modes. For the narrowband PHY, the number of sequences has been doubled to achieve an estimation gain of 6 dB compared to one symbol only. The motivation for this was to attain a relatively low performance loss due to imperfect channel estimation with the simplest technique. At the same time, a higher preamble loss can be avoided with the frame aggregation technique. Cyclic prefix length has been chosen to twice the regular guard time. This provision alleviates the required accuracy of coarse synchronization.

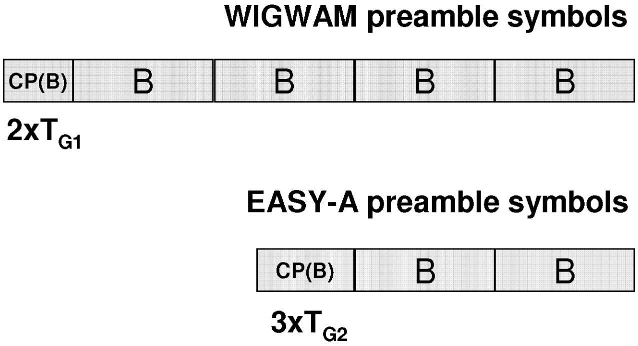


Figure 50: Inverted preamble structure

The received symbol on subcarrier l and OFDM symbol k shall be denoted as $S_{k,l}$. Symbols are indexed in offset-order, $-N_{\mathrm{FFT}}/2 \leq l < N_{\mathrm{FFT}}/2$. We assume a static channel and identical reference symbols. In case of perfect orthogonality, no phase noise and CFO, each symbol $S_{k,l}$ is given to

$$S_{k,l} = H_l R_l + \eta_{k,l} \tag{136}$$

where R_l is the reference symbol for subcarrier l and $\eta_{k,l}$ the additive noise component with power σ_η^2. The magnitude $|R_l|$ is chosen to be constant to obtain a constant average estimation quality for all subcarriers. This was achieved with BPSK reference symbols $R_l \in \{-1, 1\}$, as discussed in Section 5.6. We generalize this case by merely assuming $|R_l| = 1$. To obtain estimates $\widetilde{H}_l$ of the true channel coefficients H_l using N_{ref} reference OFDM symbols, the WIGWAM receiver calculates

$$\widetilde{H}_l = \overline{R_l} \cdot \frac{1}{N_{\mathrm{ref}}} \cdot \sum_{k=1}^{N_{\mathrm{ref}}} S_{k,l} = \overline{R_l} \cdot \frac{1}{N_{\mathrm{ref}}} \cdot \sum_{k=1}^{N_{\mathrm{ref}}} (H_l R_l + \eta_{k,l}) = H_l + \left(\frac{1}{N_{\mathrm{ref}}} \cdot \sum_{k=1}^{N_{\mathrm{ref}}} \eta_{k,l} \right) \tag{137}$$

In essence, the receiver performs a DFT of the channel response in an indirect fashion. Since the noise components are uncorrelated, noise power is reduced to $\sigma_\eta^2/N_{\mathrm{ref}}$, and $N_{\mathrm{ref}} = 4$ leads to the mentioned 6 dB estimation gain. In practice, the full gain is not achieved in presence of phase noise and residual CFO.

For the wideband mode, we have to consider shorter frames for a standard payload of 2048 bytes, as mandated by TG3c. For this reason, the preamble length was reduced to two symbols to yield less overhead (Figure 50). On the other hand, the cyclic prefix was set to three times the regular guard time to avoid ISI in any case. We note that for a properly dimensioned OFDM system, the majority of the occurring channel responses will be oversampled in frequency domain with the presented standard channel estimation method.

To illustrate this point, let's suppose that all N_{FFT} subcarriers were used. In this case, the DFT could analyze channels with a channel response length of up to N_{FFT} time-domain samples, provided that the guard time is also extended to N_{FFT} samples. But since the channel impulse response is assumed to be shorter than the regular guard time, which itself is about 25% of the DFT length or less, the channel is oversampled. Hence, low-pass filtering in frequency domain can be applied for noise reduction, having the character to smooth the channel frequency response. It is well known that cyclic convolution in time domain is equivalent to multiplication in frequency domain ([Joh96]), and the opposite (dual) relationship is also true. The low-pass filter operating in frequency domain attenuates delay components of the channel impulse response far away from the mean excess delay, where most of the energy is concentrated[1].

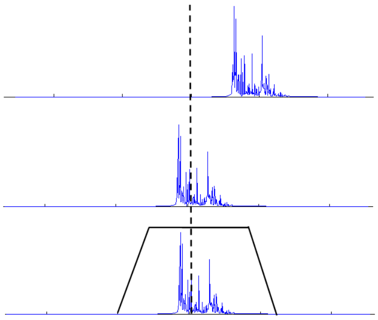


Figure 51: Timing adjustment and noise-reduction filtering

This filter can only be efficiently applied after the channel has been shifted in time domain, so that the mean excess delay (as defined in Equation (243) in Appendix B.5) is made zero. This principle is shown in Figure 51. The mean excess delay τ_{mean} is estimated by the fine time synchronizer[2], Section 5.10. Exploiting the time-shifting property of the DFT, a shift of the channel impulse response by $-\tau_{\mathrm{mean}}$ as shown in the figure can be realized by a phase rotation of the channel coefficients $\widetilde{H}_l$ with a factor $\exp(j2\pi\tau_{\mathrm{mean}} l/(TN_{\mathrm{FFT}}))$, where T denotes the sample duration.

$\mathcal{M}_q$ shall be the set of subcarrier indices defined as follows: Subcarrier l belongs to $\mathcal{M}_q$, if this subcarrier is surrounded by $2q$ active neighbor subcarriers with indices $l-1, l-2, ..., l-q$ and $l+1, l+2, ..., l+q$. Then for all $l \in \mathcal{M}_q$, the initially attained coefficients $\widetilde{H}_l$ are replaced with smoothed coefficients $\widehat{H}_l$.

$$\widehat{H}_l := \sum_{n=-q}^{q} \mathcal{B}_n \widetilde{H}_{l-n} \exp(j2\pi\tau_{\mathrm{mean}}(l-n)/(TN_{\mathrm{FFT}})) \tag{138}$$

where $\mathcal{B}_n$, $-q \leq n \leq q$, are the coefficients of a $(2q+1)$-tap symmetrical low-pass filter. q subcarriers at the left and another q at the right edge of the signal spectrum and the same amount of subcarriers at

---

[1] An optimum method would be to apply a Wiener filter.

[2] To avoid confusion, we note that the fine time synchronizer actually estimates $-\tau_{\mathrm{mean}}$.

Table 9: Coefficients of smoothing filter

tap	value	tap	value
-6	0.004	1	0.284
-5	0	2	0.072
-4	-0.024	3	-0.034
-3	-0.034	4	-0.024
-2	0.072	5	0
-1	0.284	6	0.004
0	0.400		

DC do not have enough q neighbors at each side[1]. These are left unaffected.

$$l \notin \mathcal{M}_q \Rightarrow \widehat{H}_l := \widetilde{H}_l \exp(j2\pi\tau_{\mathrm{mean}} l/(TN_{\mathrm{FFT}})) \tag{139}$$

The whole operation consisting of (138) and (139) may be written in a (sloppy) shorthand notation as

$$\widehat{H}_l := \mathcal{B}_n \otimes \left[\widetilde{H}_l \exp(j2\pi\tau_{\mathrm{mean}} l/(TN_{\mathrm{FFT}}))\right] \tag{140}$$

For the wideband system, a 13-tap filter has been introduced with coefficients given in table 9 and a shaping curve shown in Figure 52. A "pass band" of $0.4 \cdot N_{\mathrm{FFT}} = 410$ samples = 190 ns is left unaffected. The graph in Figure 53 is a plot of the estimation SNR with and without this filter, for the NLOS residential model CM2.3[2]. The additional estimation gain provided with smoothing is about 4.5 dB. Estimation performance for all other channels including RF impairments is given in Section 5.12.

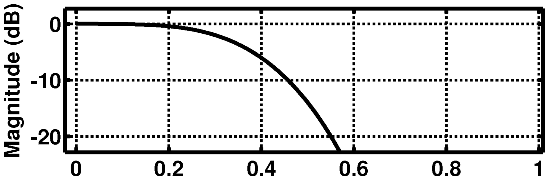


Figure 52: Smoothing filter response

Beside the channel coefficient estimates $\widetilde{H}_l$, subcarrier power levels are needed for soft-bit metrics and pilot-aided phase estimation (tracking). These power coefficients $\widehat{P}_l$ have been simply estimated according to Equation (141). The power estimates are not unbiased, but still give acceptable performance.

$$\widehat{P}_l := |\widehat{H}_l|^2 \tag{141}$$

---

[1] At DC, zero or virtual subcarriers are inserted to avoid DC offset problems.

[2] The plot relates input to output SNR. Input SNR is measured in the DFT bandwidth (= channel bandwidth). Without filter, the estimation gain is 4 dB, made up by 3 dB obtained with averaging and another 1 dB, because signal bandwidth is about 80% of the DFT bandwidth.

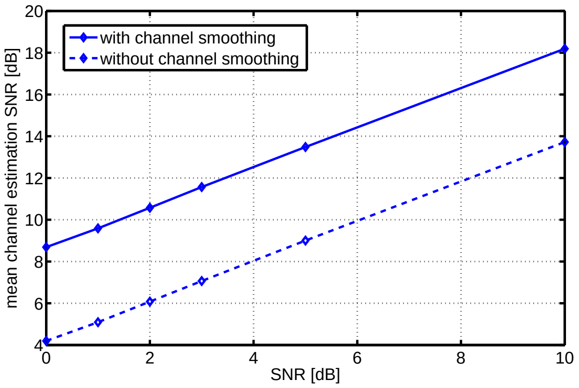


Figure 53: Channel estimation SNR with and without the smoothing filter (as published in [Max09a])

5.9.2 Equalization

Equalization can be performed in various ways following different criterions. The two considered equalization schemes assume perfect channel state information. The "zero-forcing" equalizer (ZF-EQ) simply performs channel inversion. This means that the received data or pilot subcarrier symbols $S_{k,l}^{(\mathrm{rx})}$ for data OFDM symbol k and subcarrier l are corrected according to (142). The channel estimation is assumed to be unbiased, and the same is true for the equalized symbols $\widehat{S}_{k,l}^{(\mathrm{rx})}$.

$$\widehat{S}_{k,l}^{(\mathrm{rx})} = S_{k,l}^{(\mathrm{rx})} \cdot A_l \tag{142}$$

$$A_l = \frac{1}{H_l} \tag{143}$$

The minimum-mean-square equalizer (MMSE) minimizes the variance of the Euclidean distance between the transmitted and received constellation symbols avoiding heavy noise amplification. The equalization factor is given to ([Ahm99])

$$A_l = \frac{1}{H_l} \cdot \frac{1}{1 + 1/\mathrm{SNR}_l} \tag{144}$$

where SNR_l is the SNR for the l-th subcarrier. This equalizer will attenuate the symbols for subcarriers with high noise floor. In this way, it will produce biased results with QAM constellation points deviating from the ideal positions. According to ([Ahm99]), it does not minimize the bit error rate. For this reason, MMSE was not used for this work. Note that for the soft-bit metrics described in Appendix D.3, which are based on the maximum-likelihood function, the zero-forcing EQ is assumed. Hence, there is no need for an Euclidean distance minimization when dealing with bit metrics.

On the other hand, we could think of using the MMSE to equalize the pilot subcarriers. Instead, pilot power levels (Equation (141)) have been well incorporated in the tracking algorithms to include reliability information. Therefore, the zero-forcing scheme can be successfully applied for data and pilot subcarriers.

Next, we consider the real case of imperfect channel coefficient estimates. The estimated channel coefficients $\widehat{H}_l$ will differ from the real coefficients by ΔH_l, $\widehat{H}_l = H_l + \Delta H_l$. At first, the received symbols are written as

$$S_{k,l}^{(\mathrm{rx})} = H_l S_{k,l}^{(\mathrm{tx})} + \eta_{k,l} \tag{145}$$

For the ZF-EQ, the equalized symbols read

$$\widehat{S}_{k,l}^{(\mathrm{rx})} = \frac{S_{k,l}^{(\mathrm{rx})}}{H_l + \Delta H_l} = \frac{H_l S_{k,l}^{(\mathrm{tx})}}{H_l + \Delta H_l} + \frac{\eta_{k,l}}{H_l + \Delta H_l} = S_{k,l}^{(\mathrm{tx})} - \frac{\Delta H_l}{H_l + \Delta H_l} S_{k,l}^{(\mathrm{tx})} + \frac{\eta_{k,l}}{H_l + \Delta H_l} \tag{146}$$

Carriers in deep fade can experience arbitrary large symbol errors after equalization, since the denominator can approach zero. In a practical solution, equalization will clip outside a given range of numbers. The power weighting factors $\widehat{P}_l$ will ensure in a majority of cases that errors are kept under control. The performance of the whole scheme is covered in Section 5.12.

5.10 Fine time synchronization

The goal of fine time synchronization is to minimize the intersymbol interference by choosing the best position for the DFT window. The problem was addressed in Section 4.3.1. The best result is obtained if the receiver would have perfect knowledge of the channel impulse response $h(m)$ in time domain. Then it can choose the DFT window start position based on the maximization of Equation (66). To simplify notations, we consider one symbol transmitted at n=0 and convolved with the channel $h(m)$. Combining Equations (58) and (66), we obtain the best position K_{opt} for the DFT window for this particular symbol as

$$K_{\mathrm{opt}} = N_g + \arg\max_k \left\{ \sum_{m=k}^{k+N_g} |h(m)|^2 \right\} \tag{147}$$

N_g denotes the guard interval length. To evaluate the precision of any fine-synchronization scheme, we use Equations (58), (59) and (65) to estimate the created interference power σ_I^2 for a chosen DFT window position K.

$$\sigma_I^2 = 2P_{\mathrm{tx}} \cdot \left[\sum_{m=-\infty}^{\infty} |h(m)|^2 \cdot N_E(n)/N_s \right] \tag{148}$$

$$N_E(n) = \max\left\{\max(0, m-K), \max(0, K-N_g-m)\right\}$$

We assume that minimization of (148) is approximately equal to maximization of (147). We then can use (147) for a *genius synchronizer* in order to compare the performance of any real synchronizer against it.

For the beginning, we shortly consider a direct method to estimate the channel impulse response. This could be done as follows. At first, the channel coefficients are estimated in frequency domain. The next step consists in IDFT operation to obtain $h(m)$. Unfortunately, there are some subcarriers missing for a complete IDFT. These are the guard subcarriers at the band edge and at DC. Nevertheless, solutions to circumvent this problem might exist. In addition, application of IDFT operation is paid with additional latency. For these reasons, this method is not further considered.

A common solution is to perform cross-correlation over the second part of the preamble. Note that precursors can appear prior to the highest peak in $|h(n)|$. Therefore, if the DFT window for the data symbols would start at the maximum of $|h(n)|$, the system may experience considerable interference arising from precursor paths. This means that the cross-correlation scheme should be accurate enough to identify precursors as well. A suboptimum but practical solution could consist in starting the DFT some fixed amount of samples prior to the highest detected peak in $|h(n)|$. Cross-correlation based fine synchronization is paid with considerable computational effort.

Another method which comes at very low hardware complexity is to approximate the excess delay τ_{mean} of a particular channel response with the *mean phase increase* between adjacent subcarriers. This kind of time estimation approach in frequency domain has been rediscovered by the author, but was published before in [J. 03a], where Granado presented a post-FFT frequency and timing estimation scheme. The method has been chosen in this work due to its acceptable performance at very low complexity. The estimator essentially performs some sort of *phase unwrapping*. This could be done in the vector- as well as in the phase domain. System simulations done by the author have shown better performance for the phase domain estimator. The estimator is written as follows.

$$\triangle n_{\text{est}} := \left(\sum_{l \in \mathcal{Q}_p} [\widehat{\phi}_{l+\Delta l} - \widehat{\phi}_l]_{2\pi} \right) \cdot \alpha/\pi \tag{149}$$

$$\widehat{\phi}_l = \angle \widehat{H}_l \quad \alpha = N_{\text{FFT}}/(2\Delta l N_p) \tag{150}$$

In (149), $[\phi]_{2\pi}$ denotes the complementary modulo function mapping ϕ into the range $[-\pi, \pi)$ to attain a wrap-around behavior of the phase differences in the summation. We define this function as follows.

$$\begin{aligned} & y > 0, k \in \mathbb{Z} \\ & z = [x]_y \Leftrightarrow z = x + k \cdot y \wedge -y/2 \leq z < y/2 \end{aligned} \tag{151}$$

$\widehat{\phi}_l \in [-\pi, \pi)$ are the angles of the channel estimates $\widehat{H}_l$. $\mathcal{Q}_p$ is the set of N_p indices l, for which subcarrier l and $l + \Delta l$ are both used in the preamble (and in general). Essentially, this estimator calculates the average phase increase from all active subcarrier pairs $(l, l + \Delta l)$. The simple scheme is best understood for a flat (Dirac) channel given with continuous-time response $h(\tau) = A\delta(\tau - n_d T)$, i.e. when the signal waveform is received as one (phase shifted) copy in its original form. Without noise, channel coefficients $\widehat{H_l}$ are equal to $A \cdot \exp(-j2\pi l n_d/N_{\text{FFT}})$. The phase slope is proportional to the (non-integer) delay n_d, and the estimator (149) produces the desired output of

$$\triangle n_{\text{est}} = \left(\sum_{l \in \mathcal{Q}_p} [\widehat{\phi}_{l+\Delta l} - \widehat{\phi}_l]_{2\pi} \right) \cdot \alpha/\pi = N_p \cdot (-2\pi \Delta l n_d/N_{\text{FFT}}) \cdot N_{\text{FFT}}/(2\pi \Delta l N_p) = -n_d \tag{152}$$

For the maximum delay, which is correctly resolved, the phase differences in (149) are equal to $\pm\pi$. Hence, the delay n_d is correctly resolved if $n_d \in [-N_{\text{FFT}}/(2\Delta l), +N_{\text{FFT}}/(2\Delta l))$. Figure 54 illustrates the flat channel case. The phase unwrapper estimates the old DFT position of the receiver with respect to the preamble symbol. This position had been set by the coarse synchronizer and need to start in the cyclic prefix range[1].

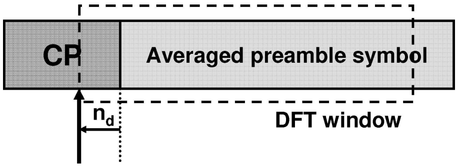


Figure 54: DFT window prior to timing adjustment for flat channel

For a multipath channel, the estimated position will approximately relate to some reference point not far from the mean excess delay of the channel. The DFT window has to be readjusted for the data symbols. We follow a heuristic scheme by simply choosing a new timing position n_{new} with a fixed offset n_{offset} earlier to the estimated reference point $n_{\text{old}} - \Delta n_{\text{est}}$, i.e. the new position is chosen as

$$n_{\text{new}} = (n_{\text{old}} - \Delta n_{\text{est}} - n_{\text{offset}})_{\text{int}} \tag{153}$$

where $(...)_{\text{int}}$ denotes rounding operation. The offset n_{offset} reduces the effective guard duration for channels with a strong LOS path followed by the decaying power profile of the dispersion. On the other hand, the offset is needed to account for precursors which are especially strong for NLOS channels.

Figure 55 shows simulation results for different offsets n_{offset} and TG3c channel models CM12, CM22, CM23 and CM32. We recall from Section 3.3 that CM12 is nearly an AWGN channel with very low delay spread. CM22 and CM23 are the residential NLOS channel for 60 and 30 degree antenna beam width and CM3.2 is the LOS office channel for 60 degree antennas having very large rms delay spread (Figure 13). The lowest possible interference levels produced by the genius synchronizer are also plotted. Apparently, an offset value of $n_{\text{offset}} = 70$ gives the best average performance taken all four channel models into account. Performance in CM1.2 and CM2.2 is not critical. For CM2.3, 90% of the channels produce an interference level not higher than 30 dB below the average signal level, and 95% stay below a level of -22.5 dB. For CM3.2, the most critical channel, there is a penalty of around 2 dB compared to the genius synchronizer. We note that channel CM3.2 might not be the most realistic model. $n_{\text{offset}} = 70$ has been chosen for the fine synchronizer, at the price of 27% of the guard time or 32 ns.

Finally, we shortly consider parameter Δl. *Noise performance* calls for larger values and *tolerance range* for smaller values of Δl. In this work, $\Delta l = 1$ has been implemented. The coarse synchronizer could have timing errors as high as half of the DFT range, but this range is not fully covered by the guard time. Hence, $\Delta l = 2$ may give about 3 dB better noise performance at an acceptable tolerance range.

5.11 Tracking of phase and timing and channel re-estimation

The receiver needs to track carrier phase and timing during frame reception. These parameters change much faster compared to the channel for any realistic indoor scenario in the home or office environment.

---

[1] To avoid ISI due to initial synchronization errors, the guard time is well extended.

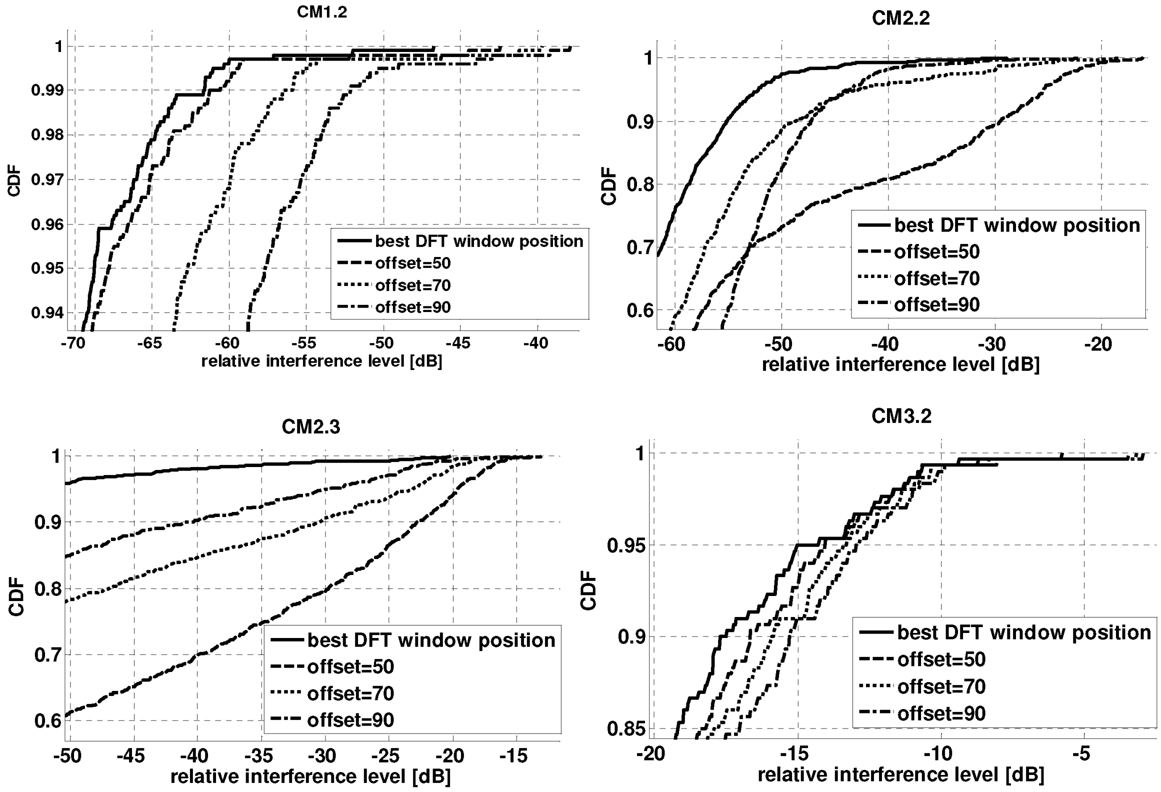


Figure 55: Fine synchronization performance for TG3c channels without noise

Therefore, a scheme where timing tracking is covered by channel tracking has not been considered. The carrier phase seen at the receiver side changes over time due to residual carrier frequency offset and phase noise. Deviation of transmitter and receiver clock as well as clock jitter are responsible for continuous timing shift.

A clean method for clock drift compensation would involve sample-wise interpolation in time domain. On the other hand, block-wise compensation of timing in frequency domain involves merely a phase rotation of subcarriers, and this task can be covered by the equalizer block. It has been calculated in Section 4.3.3 that the degradation for an ultimate clock deviation of 50 ppm is still acceptable.

Residual carrier frequency offset (CFO) and phase noise lead to thermal noise-like ICI [1]and a phase error common to all subcarriers. Phase noise could me mitigated to some extend with the aid of *iterative decoding techniques* as shown in [Ger05], but this option is hardly feasible for the high-rate, high-latency hardware architecture. Therefore, this system is restricted to perform only the usual common phase error (CPE) correction.

A tracking scheme to reduce the residual carrier frequency offset has not been investigated. Since the pilots constitute only a fraction of the received power, the estimated common phase error experiences noise-induced fluctuations and is also perturbed by phase noise. In addition, the feedback path of a CFO tracking loop would have a very high latency, since the NCO is positioned prior to the FFT. For

---

[1] The additional noise term caused by the ICI of phase noise is *not* Gaussian distributed as shown in [DP04].

this reason, the update factor of such a loop would have to be set to very small values, strongly limiting the efficiency.

5.11.1 Tracking scheme for narrowband system (WIGWAM demonstrator)

In this receiver, timing and phase are estimated and corrected in frequency domain using embedded continuous pilots. In the following, we use indexing for data and pilot subcarriers in offset notation, so that indices range from $-N_{\mathrm{FFT}}/2$ up to $+N_{\mathrm{FFT}}/2-1$. $T=1/f_T$ denotes the sample duration. We may assume that the channel is approximately constant for the considered period of time. Due to the cyclic shifting property of the DFT, the accumulated timing deviation $\Delta\tau(n)$ of the DFT window applied on the n-th OFDM symbol leads to a phase rotation $\Delta\theta(n)\cdot k$ of the symbol in subcarrier k, which is proportional to the subcarrier index (or frequency). The phase slope $\Delta\theta(n)$ is given to

$$\Delta\theta(n)=2\pi\Delta\tau(n)/(TN_{\mathrm{FFT}}) \tag{154}$$

This timing deviation is related to the reference position, for which the initially calculated channel coefficients apply. With the common phase error $\theta_0(n)$, the combined unwrapped phase error for subcarrier k in OFDM symbol n is given as

$$\theta(n,k)=\theta_0(n)+\Delta\theta(n)\cdot k \tag{155}$$

$\theta_0(n)$ and $\Delta\theta(n)$ are the unknown parameters to be estimated. Hence, the problem has the character of a line-fitting procedure in the *unwrapped* phase domain[2]. But estimation can also be performed in vector domain. The N_p transmitted pilots in the n-th OFDM symbol carry symbols denoted as $S_{n,l}^{(p)}$, $l=1...N_p$. A BPSK-modulated pseudo-random bit sequence [3] is used to create these symbols $S_{n,l}^{(p)}\in\{-1,1\}$. Pilot subcarriers are located on fixed positions $q(l)$ of a grid with constant step size $q(l+1)-q(l)=\Delta_p$. The receiver evaluates pilot symbols $Z_{n,l}^{(p)}$ after equalization and BPSK demodulation. This demodulation is carried out like descrambling by multiplication with $S_{n,l}^{(p)}$. We write $\tilde{Z}_{n,l}^{(p)}$ for the received pilot symbols prior to equalization. With estimated channel coefficients $\widehat{H}_{q(l)}=\widehat{A}_{q(l)}\exp(j\widehat{\phi}_{q(l)})$, the equalized and demodulated (sign-flipped) pilot symbols are given to

$$Z_{n,l}^{(p)}=\frac{S_{n,l}^{(p)}\tilde{Z}_{n,l}^{(p)}}{\widehat{A}_{q(l)}}\exp(-j\widehat{\phi}_{q(l)}) \tag{156}$$

In case of nearly perfect channel estimation, these pilot symbols read

$$Z_{n,l}^{(p)}\approx\exp\left(j[\theta_0(n)+\Delta\theta(n)\cdot l]\right)+\eta_{n,q(l)}/\widehat{A}_{q(l)} \tag{157}$$

where $\eta_{n,l}$ denote the AWGN (and ICI) noise contributions in the pilot subcarriers. In the strict sense, these noise components have different variances (power levels) $\sigma^2_{\eta,q(l)}$. The reason is that the created

---

[2] The unwrapped phase is naturally expanded to the full number range.

[3] This is done to avoid spectral lines and increases robustness against IQ mismatch. In this way, pilot subcarriers produce the same spectrum as all other subcarriers carrying random data.

ICI from phase noise depends on the received level of the neighbor subcarriers. The neighbors produce most of the ICI for the considered subcarrier and vary in power for frequency selective channels. For simplicity, we will assume that ICI is small and neglect the noise variations, i.e. we assume an equal noise power of σ_η^2 for all subcarriers prior to equalization. Note that without noise, timing and phase deviations, the pilot symbols $Z_{n,l}^{(p)}$ would all point in positive direction of the real axis.

In the search for a good estimator, it was followed a rather intuitive approach without the attempt to find the minimum-variance unbiased (MVU) estimator ([Ste93]). Estimation of the phase slope $\Delta\theta(n)$ can be done independently from the CPE. We incorporate reliability information given by the pilot power levels $\widehat{P}_{q(l)} = (\widehat{A}_{q(l)})^2$ and make the ansatz

$$\Delta\widehat{\theta}(n, \Delta_l) := \frac{1}{\Delta_p \Delta_l} \angle \left\{ \sum_{l=1}^{N_p - \Delta_l} \left[\overline{Z_{n,l}^{(p)}} Z_{n,l+\Delta_l}^{(p)} \right] \cdot \gamma_p(P_{q(l)}, P_{q(l+\Delta_l)}) \right\} \tag{158}$$

This estimator essentially evaluates the phase difference between every pilot pair $(l, l + \Delta_l)$ by calculating the product $\mathcal{X}_l := [\overline{Z_{n,l}^{(p)}} Z_{n,l+\Delta_l}^{(p)}]$. The function $\gamma_p(P_1, P_2)$ creates a weighting factor for each product. Since the denominator $\Delta_p \Delta_l$ is the index difference between the related subcarriers, Equation (158) performs estimation of the phase slope $\Delta\theta(n)$ as required. Because estimation is based on data from one OFDM symbol, we call (158) the *instantaneous phase slope estimate*. Δ_l is a free parameter. Higher values lead to better noise performance. However, the maximum resolvable timing deviation as a function of Δ_l is given to

$$\Delta\tau_{\max}(\Delta_l) = \frac{T \cdot N_{\mathrm{FFT}}}{2\Delta_p \Delta_l} \tag{159}$$

For good estimation performance, we may choose $\gamma_p(P_1, P_2)$ to mimic *maximum-ratio combining* ([D.G03]). Since pilots are evaluated *after* equalization, and we have assumed a flat noise floor, each product term should be weighted with the inverse of the effective noise power or equivalently, with the effective SNR. We approximate each of the products $\overline{z_1} z_2$ of two noisy complex numbers $z_1 = c_1 + \eta_1$, $z_2 = c_2 + \eta_2$ as $\overline{z_1} z_2 = \overline{c_1} c_2 + \overline{c_1} \eta_2 + \overline{\eta_1} c_2 + \overline{\eta_1} \eta_2 \approx \overline{c_1} c_2 + \overline{c_1} \eta_2 + \overline{\eta_1} c_2$. Since the pilots have a level of unity after equalization, the inverse of noise power in each product term $\mathcal{X}_l$ is approximately equal to

$$\frac{1}{\sigma_{\mathcal{X}_l}^2} \approx \frac{1}{\sigma_\eta^2 / \widehat{A}_{q(l)}^2 + \sigma_\eta^2 / \widehat{A}_{q(l+\Delta l)}^2} = \frac{1}{\sigma_\eta^2} \cdot \frac{1}{1/\widehat{P}_{q(l)} + 1/\widehat{P}_{q(l+\Delta l)}} \tag{160}$$

Therefore, we may define $\gamma_p(P_1, P_2)$ as

$$\gamma_p(P_1, P_2) := \frac{1}{1/P_1 + 1/P_2} \tag{161}$$

The weaker subcarrier dominates the noise floor and therefore the weighting factor. For equal subcarrier levels, $\gamma(P, P)$ will be reduced by 3 dB compared to $\gamma(P, \infty)$. For the implementation, a simpler weighting rule has been applied.

$$\widetilde{\gamma}_p(P_1, P_2) := \min(P_1, P_2) \tag{162}$$

Again the weaker subcarrier sets the weighting factor, but as subcarrier levels become equal, this weighting factor will be higher. This rule gives only little degradation, but greatly simplifies implementation.

In order to correct the CPE, pilots are first rotated to compensate the timing-related phase errors. The used phase slope estimate is written as $\Delta\widetilde{\theta}(n)$ and will differ from the instantaneous estimate, as discussed later. The phase-rotated pilots $\mathcal{Z}_{n,l}^{(p)}$ are given to

$$\mathcal{Z}_{n,l}^{(p)} = Z_{n,l}^{(p)} \exp(-j\Delta\widetilde{\theta}(n)q(l)) \tag{163}$$

Without noise, all pilot would point in direction of the CPE. In order to apply maximum-ratio combining, the CPE is obtained as the angle of the power-weighted vector sum. Estimation in vector domain avoids the problem of phase ambiguity for $\theta_0(n)$ in the vicinity of $\pm\pi$.

$$\widehat{\theta}_0(n) = \angle \left\{ \sum_{l=1}^{N_p} \mathcal{Z}_{n,l}^{(p)} P_{q(l)} \right\} \tag{164}$$

Now we come back to timing estimation. The scheme incorporated in the narrowband PHY implementation is based on the instantaneous estimate (158) with $\Delta_l = 1$. This setting tolerates the highest timing offset. With a pilot distance of $\Delta_p = 14$, the maximum sample deviation is equal to $\Delta n_{\max} = \Delta\tau_{\max}(\Delta_l = 1)/T = 256/(2 \cdot 14) = 9.14$ samples. There is no DFT window readjustment incorporated so that the maximum frame length is limited by $\Delta n_{\max}$ and the tolerable clock deviation, which shall be set to ϵ = 100 ppm = 10^{-4}. N_{sym} denotes the symbol duration, equal to $N_{\text{sym}} = 320$ samples. The maximum number of OFDM symbols is obtained as $N_{\max} = \Delta n_{\max}/(N_{\text{sym}} \cdot \epsilon) \approx 285$ symbols. Leaving a margin of 10% for the phase slope to account for noise-induced variations gives a maximum frame length of around 257 OFDM symbols. With 192 data subcarriers and BPSK modulation using rate-1/2 encoding, a smaller packet size of 172 OFDM symbols is needed. With T=2.5 ns, timing can drift 15 ns at most. This drift can cause increased ISI. This ISI is tolerated for the narrowband system, since BPSK-1/2 is robust and higher modes require shorter frame lengths.

Simulations have shown that the variance of the estimated phase slope is too high and causes performance loss, see Section 5.12. This loss appears as link quality degradation from inner to outer subcarriers located at higher frequencies. Since the CPE correction depends on the estimated phase slope, correction accuracy of the CPE is also degraded. Therefore, an averaging method over many OFDM symbols has been applied to improve accuracy. We assume a fixed frequency deviation between transmitter and receiver clock and make use of an adaptive predictor to derive an averaged phase slope $\Delta\bar{\theta}(n)$ from the instantaneous estimate $\Delta\widehat{\theta}(n, \Delta_l)$. This averaged phase slope is finally used in (163) with $\Delta\widetilde{\theta}(n) := \Delta\bar{\theta}(n)$. In the following, we drop the parameter Δ_l for convenience. The predictor is given as

$$\Delta\bar{\theta}(n) = [1 - \gamma_f(n)] \cdot [\Delta\bar{\theta}(n-1) + \Delta\Delta\bar{\theta}(n-1)] + \gamma_f(n) \cdot \Delta\widehat{\theta}(n) \tag{165}$$

$$\Delta\Delta\bar{\theta}(n) = [1 - \beta_f] \cdot \Delta\Delta\bar{\theta}(n-1) + \beta_f \cdot [\Delta\widehat{\theta}(n) - \Delta\widehat{\theta}(n-1)] \tag{166}$$

(166) performs continuous estimation of $\Delta\Delta\bar{\theta}(n)$. This parameter constitutes the increase of the phase slope from symbol to symbol and is directly proportional to the clock deviation. Estimation is controlled

with an update- or 'forgetting'-factor β_f. (165) is used for estimation of the averaged phase slope. The output is a weighted sum of the instantaneous estimate and the predicted estimate, with $\gamma_f(n)$ as the weighting factor. Phase correction starts for $n = 1$, the signal field, and continues for data symbols. If no *a priori* information is used, the initial conditions are $\Delta\bar{\theta}(n-1) = 0$ and $\Delta\Delta\bar{\theta}(n) = 0$. Therefore, the predicted phase slope is *biased*, because the initial assumption is to have no clock deviation. To reduce the error and settling time but not sacrifice noise performance too much, the update factor $\gamma_f(n)$ is chosen to $\gamma_f(n) = 0.2$ for n=1...20 and $\gamma_f(n) = 0.1$ for $n > 20$. The presented scheme has been implemented this way in the (narrowband) WIGWAM demonstrator and published in [Max09b]. The simulated system performance is given in Section 5.12.

5.11.2 Improved tracking scheme for wideband system

For the wideband mode with an DFT size of 1024 and a bandwidth of 2160 MHz, the solution described above fails to work properly. For a fixed subcarrier spacing, the variance of the phase error due to timing deviation increases with the order of $\sim N^2$. If the fraction of pilots is kept constant, the accuracy improvement of the instantaneous phase slope estimate for $\Delta_l = 1$ seems to be only in the order of[1] $\sim N$. However, the system can be improved. At first, we can overcome the restriction of the frame length with a modified scheme. Since the receiver continuously tracks the progressing phase slope, phase discontinuities at $\pm\pi$ can be prevented by extending the number range. The instantaneous estimate is not immediately normalized by $1/(\Delta_p\Delta_l)$ to prevent ambiguity. Equation (158) is replaced with

$$\vartheta(n, \Delta_l) = \angle\{ \sum_{l=1}^{N_p-\Delta_l} \left[\overline{Z_{n,l}^{(p)}} Z_{n,l+\Delta_l}^{(p)} \right] \cdot \gamma_p(P_{q(l)}, P_{q(l+\Delta_l)}) \} \tag{167}$$

$\vartheta(n)$ will not differ much from the averaged value $\overline{\vartheta}(n-1)$ from the previous iteration. Therefore, we unwrap the phase $\vartheta(n)$ into the correct range via

$$\vartheta_u(n) = \bar{\vartheta}(n-1) + [\vartheta(n) - [\bar{\vartheta}(n-1)]_{2\pi}]_{2\pi} \tag{168}$$

In (168), $[x]_y$ denotes the complementary-modulo function as defined in Equation (151). We let the predictor run over $\vartheta(n)$ and $\vartheta_u(n)$ as follows.

$$\bar{\vartheta}(n) = [1 - \gamma_f(n)] \cdot [\bar{\vartheta}(n-1) + \Delta\bar{\vartheta}(n-1)] + \gamma_f(n) \cdot \vartheta_u(n) \tag{169}$$

$$\Delta\bar{\vartheta}(n) = [1 - \beta_f] \cdot \Delta\bar{\vartheta}(n-1) + \beta_f \cdot [\vartheta(n) - \vartheta(n-1)]_{2\pi}$$

$$\Delta\widetilde{\theta}(n) := \bar{\vartheta}(n)/(\Delta_p\Delta_l) \tag{170}$$

The modified scheme allows one to choose higher values for Δ_l and improve noise performance. We assume a flat AWGN channel as the result of averaging over all appearing channel responses. We define the estimation gain as $G = \sigma^2_{\text{basic}}/\sigma^2_{\text{combined}}$, where σ^2_{basic} is the variance of the estimated phase slope

---

[1] This statement is not exactly true, because all but two pilots are used in two adjacent pilot pairs. Hence, the different estimations have dependencies. We have seen in Section 4.4.2 that such dependencies can have a beneficial effect for the combined estimation. But due to the weighting, behavior is different for this estimator. Analysis is involved and beyond the scope of this text.

based on one pilot-pair of adjacent pilots ($\Delta_l = 1$) and $\sigma^2_{\mathrm{combined}}$ the variance using all pilot pairs with some value for Δ_l. If we choose $\Delta_l \geq N_p/2$, every pilot appears only once in the estimation. For a flat channel, the estimation gain is equal to $G = {\Delta_l}^2 \cdot (N_p - \Delta_l)$, which is maximized for $\Delta_l = (2/3)N_p$ to $G = (4/27)N_p^3$. Hence, with rising number of subcarriers, the accuracy improvement scales at least as required. Furthermore, simulation shows that the accuracy for $\Delta_l = (2/3)N_p$ or even $\Delta_l = N_p/2$ is sufficiently high without the predictor. After all, the originally introduced prediction filter can be avoided. Only the unwrapping logic given by Equation (168) needs to be retained. Omitting the predictor also alleviates the integration of two additional features:

- To avoid interference caused by clock drift, the DFT window has to be readjusted based on the estimated position of the timing tracker.
- The wideband system shall support midamble-based channel re-estimation.

As previously mentioned, the transmitter can embed midambles in the data frame for channel re-estimation. The general frame structure was shown in Figure 35 in Section 5.4. The midamble consists of two channel estimation symbols, which are chosen identical to the second preamble part. The only difference lies in a reduced prefix length of two instead of three regular prefix lengths in order to attain a midamble length of two regular OFDM symbols[1]. This is possible since fine frame timing is already established.

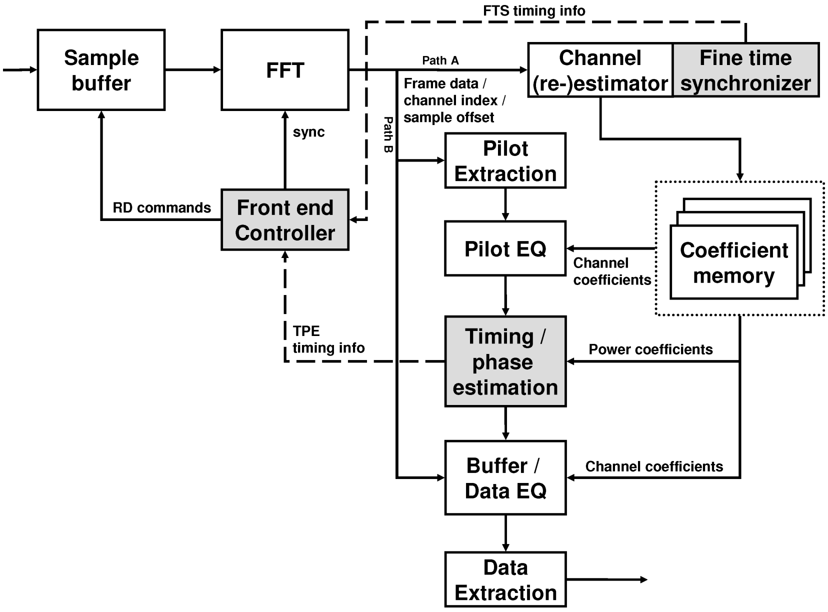


Figure 56: Structure of inner receiver for wideband PHY

At this point, it is necessary to consider the complete receiver front end, since operation of the timing tracking is affected by the channel update mechanism. System behavior can be explained with the aid of Figure 56. A state machine called *front end controller* is responsible for burst-wise FFT operations.

---

[1] In addition, the signs of the midambles are modulated by a scrambler to improve the spectral properties.

This controller initiates FFT operations at determined start positions. FFT output data consists either of the preamble or midambles, or of data symbols. Preamble and midambles are delivered to the channel estimator on path A whereas all data symbols enter the equalization path B. In this way, midamble-based channel updates require negligible circuit overhead. Frame timing is also recalculated as for the preamble.

Basically, there are two timing adjustment mechanisms working in parallel. The frame timing is readjusted by the fine frame synchronizer as well as by the timing tracker. The global controller receives timing commands from both blocks, called FTS and TPE as shown in the figure. These commands carry the sample offset to be applied with respect to the previous timing.

The situation is slightly complicated by the general latency issue. A channel/timing (FTS) update performed on a midamble prior to some data symbol N becomes effective only for OFDM symbol $N + D_{\mathrm{mid}}$ and subsequent symbols. D_{mid} denotes the channel update delay, composed by the delay of the channel estimation path and the FFT. A similar situation arises for the timing tracker. If a TPE command is initiated after observation of symbol N, new timing is applied on symbol $N + D_{\mathrm{track}}$, i.e. delayed by D_{track} symbols. This behavior has been incorporated into the MATLAB model. The delay parameters were estimated to $D_{\mathrm{mid}} = 14$ and $D_{\mathrm{track}} = 12$, prior to an existing hardware implementation.

The situation is resolved with paged memory and additional control information attached to the OFDM symbol data by the global controller. This control information consists of a *channel index* and the applied *sample offset*. The channel coefficient memory can hold three different coefficient sets, and the channel index determines which set is valid for the current symbol data. A channel update overwrites the oldest channel data in the memory and increments the channel index.

We now observe the system from the global controllers perspective. Every TPE command also carries the channel index. If the current index is newer, the TPE command is ignored. In this way, OFDM symbols which are processed with the updated channel estimation, are unaffected by an unnecessary TPE command referring to the old update. The global controller will handle an FTS timing command from the frame synchronizer first. Since this command increments the channel index, a TPE command arriving at the same time will be ignored. Therefore, an FTS has higher priority. Finally we have to note that TPE timing readjustment is prohibited between the two midamble symbols.

Now we consider the behavior of the timing tracker. It features a two-state FSM with an idle and (default) active state. In active state, the tracker can send a TPE command, but must enter the idle state afterwards. In idle state, no TPE commands can be initiated. Of course, pilot-based phase correction is performed all the time. The tracker remains idle until one of the following conditions apply.

1. The channel index has changed, i.e. a new channel update takes effect on the current symbol.

2. The sample offset is nonzero, i.e. the previous TPE takes effect on the current symbol.

The waiting mechanism ensures flawless behavior under any combination of latency parameters D_{mid} and D_{track}. The only condition to be met is that the symbol distance between two midambles is larger than the delay D_{mid}.

The (dimensionless) sample offset for the current symbol will be denoted as $\Delta s(n)$. Recall that this offset is nonzero only after a timing readjustment. In case of a channel update, it has no meaning. The final scheme for phase slope (timing offset) estimation works as follows.

1. An instantaneous phase slope estimate $\vartheta(n)$ is obtained using Equation (167) with $\Delta_l = N_p/2$.
2. The estimate $\vartheta(n)$ is unwrapped into $\vartheta_u(n)$ using a reference phase ϕ_{ref}. This reference phase is determined by $\vartheta_u(n-1)$, the sample offset $\Delta s(n)$ and the channel index.

If the channel index is the same as for the previous symbol, the reference phase is calculated with Equation (171). ϕ_{ref} is equal to the previous value $\vartheta_u(n-1)$ for $\Delta s(n) = 0$. In case of a nonzero sample offset, the reference phase is calculated for the shifted time base.

$$\phi_{\mathrm{ref}} := \vartheta_u(n-1) + 2\pi \Delta s(n) \Delta_p \Delta_l / N_{\mathrm{FFT}} \tag{171}$$

A new channel index marks the first symbol equalized with a new set of channel coefficients. Channel data and time base have changed, and (171) cannot be applied. For this case, we introduce an additional *coarse instantaneous phase slope estimate* $\vartheta_{\mathrm{coarse}}(n)$ to avoid phase ambiguity. This is just the phase slope estimate in (167) for $\Delta_l = 1$. We can use this value for phase unwrapping, since timing cannot exceed the covered range of the coarse estimator within the amount of time.

$$\phi_{\mathrm{ref}} := \Delta\widehat{\theta}(n, \Delta_l = 1) \tag{172}$$

Unwrapping of $\vartheta(n)$ and normalization are done as before, just without the prediction filter.

$$\begin{aligned} \vartheta_u(n) &:= \phi_{\mathrm{ref}} + [\vartheta(n) - [\phi_{\mathrm{ref}}]_{2\pi}]_{2\pi} \\ \Delta\widetilde{\theta}(n) &:= \vartheta_u(n)/(\Delta_p \Delta_l) \end{aligned} \tag{173}$$

TPE command policy

As previously discussed, the tracker *in active state* should send a TPE command if the FFT window has drifted away from its initial position. This is implemented as follows. After the unwrapped phase slope estimate $\vartheta_u(n)$ has been calculated, the estimated dimensionless sample drift $\Delta\widehat{s}(n)$ is given to

$$\Delta\widehat{s}(n) = \vartheta_u(n)/(2\pi) \cdot N_{\mathrm{FFT}}/(\Delta_p \Delta_l) \tag{174}$$

The adjustment of the window can be decided based on a threshold. Furthermore, the system is simplified if the adjustment is done with a fixed step size Δn, and this step size can be equal to the threshold. The policy works as follows.

1. If $\Delta\widehat{s}(n) > \Delta n$, the tracker sends a TPE command $\Delta_{\mathrm{TPE}}(n) = -\Delta n$ in order to shift the FFT window Δn samples back.
2. If $\Delta\widehat{s}(n) < -\Delta n$, the tracker sends a TPE command $\Delta_{\mathrm{TPE}}(n) = +\Delta n$ in order to shift the FFT window Δn samples ahead.

We make the following remarks:

- The sample drift does not have to be calculated according to (174). Instead of using $\Delta\widehat{s}(n)$, the hardware can use $\vartheta_u(n)$ to compare it against the corresponding threshold phase values.
- A TPE command can have only three values $+\Delta n$, $-\Delta n$ and 0, if no adjustment is needed. Accordingly, $\Delta s(n)$ will have the same three values. Both types of commands can be encoded with two bits since the main controller and tracker both know the step value.
- The latency of them system requires that Δn is high enough to avoid a race condition between the sampling drift and compensation mechanism, for the maximum expected clock deviation. For the demonstrator, a value of $\Delta n = 2$ has been proven to be sufficient.

Conclusion

In summary, this section has shown the ’evolution’ of the tracking mechanism from the solution applied to the narrowband system to the final scheme for the wideband version. Tracking affects receiver performance, which is presented in Section 5.12. The final scheme achieves good performance and eliminates the need for a prediction filter.

5.12 Receiver performance

5.12.1 Performance of narrowband PHY for HHI channels

At first, performance results for 16-QAM-1/2 modulation are given in Figure 57 as published in [Max09b]. The frame error rate was investigated for the Omni-Vivaldi LOS channel (Section 3.2) using a packet length of 2048 source bytes transmitted in 43 OFDM OFDM data symbols. All overhead losses (preamble, signal field, pilot symbols, cyclic prefix) were accounted for. System performance was compared between a real and an idealized case with no phase noise and timing drift. For the real case scenario, 30 ppm clock frequency deviation and a Wiener oscillator model for the phase noise had been assumed. Both transmitter and receiver VCO have a single sideband phase noise value of L_{SSB} = -93dBc/Hz at 1 MHz offset. The predictor-based tracking scheme for carrier phase and timing was chosen with the parameters implemented in the hardware demonstrator: $\beta = 0.1$, $\gamma(n) = 0.2$ for $n <= 20$ and $\gamma(n) = 0.1$ for $n > 20$ (Equations (165)-(166) in Section 5.11.1). A packet error rate of 10% is met at Eb/N0 of 12.3 dB. The loss in performance due to phase noise and sampling drift is about 0.7 dB.

This result is complemented by the following additional simulations in Figure 58 performed for the HHI NLOS channels in order to gain more insight into the behavior of the system. Packet size and data mode are chosen to be the same 2048 bytes transmitted with 16-QAM-1/2. As shown in Section 3.2, the NLOS channels experience much higher power fluctuations, so that performance must be degraded compared to the LOS case. In order ’to tell the full story’, we consider seven different cases which are summarized in table 5.12.1. Realistic parameters regarding power amplifier and clock jitter were given in Section 2.7. Instead of a non ideal ADC having an ENOB of 7.4 bits, we consider an ideal 7 bit ADC. Compared to Figure 57, we this time include the performance in case of no RF impairments, no preamble-based CFO correction and no tracking of carrier phase and timing. The resulting curve

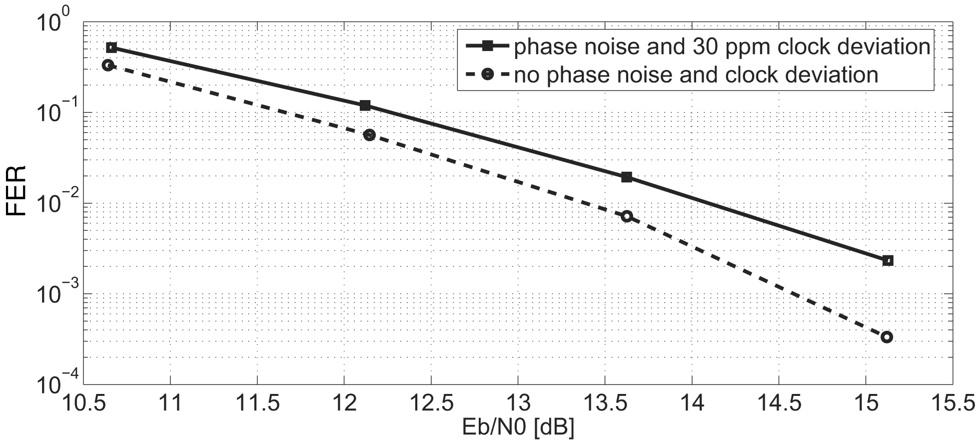


Figure 57: 480 MBit/s data mode performance in HHI Omni-Vivaldi LOS channel

(1) is taken as the reference curve. The performance measure shall be defined here as the point where a frame-error rate of 8% is achieved. This critical FER is specified in the TG3c system comparison document (citeTG3cSysComp).

Table 10: Link conditions and baseband mode

(1)	no RF impairments, no carrier frequency offset, CFO/CPE/timing correction disabled
(2)	no RF impairments, max. CFO = 2.5 MHz, CFO/CPE/timing correction enabled without timing prediction
(3)	same as in (2) with timing prediction enabled
(4)	same as in (3), but including 50 ppm clock deviation and Wiener phase noise of -90 dBc/Hz @ 1 MHz (-93 dBc/Hz at both sides)
(5)	same as in (4), but including Wiener oscillator causing clock jitter with -97 dBc/Hz @ 10 kHz
(6)	same as in (5), but including a PA Rapp model with p=1.4 driven with 5 dB IBO
(7)	same as in (6), but including ideal 7 Bit ADC quantization with 3 bit headroom

Apparently, the system shows very poor performance without the addition of the timing prediction filter (curve 2). If the prediction filter is included, the loss due to imperfect CFO estimation and tracking is reduced to about 1 to 1.2 dB at FER=8%. The overall loss is increased to 2 and 2.5 dB, respecitvely, when a worst-case clock deviation of 50 ppm and phase noise of -93 dBc/Hz @ 1 MHz offset are added for the transmitter *and* receiver VCO. We see that the influence of the specified clock jitter is negligible. Hence, the initial decision to design the predictor for a deterministic error has proven to be correct, at least for the considered clock generation. Curve (6) results from adding the power amplifier into the

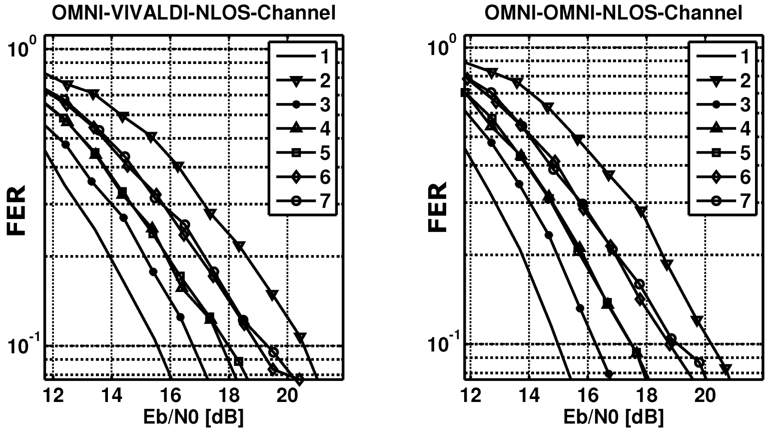


Figure 58: 480 MBit/s data mode performance for HHI NLOS channels

system. The PA is modeled with the RAPP equation, choosing $P = 1.4$. At an input power backoff of 5 dB, we observe an additional degradation of 2 dB. According to the graph in Figure 4 in Section 2.5, the output power would also drop by 1.2 dB. For this type of PA, the backoff must be chosen to higher values, at least 7 dB. The last curve includes an ideal 7-bit ADC driven at -18 dBFS (3 bit headroom). There is no additional performance degradation to be observed.

5.12.2 Synchronization performance for wideband mode

In this section, we study the performance of the synchronization scheme for the wideband PHY using channel models CM2.2, CM1.2 and CM3.2 for 60 degree transmitter and receiver antenna. Every channel is normalized to reflect the instantaneous SNR seen by the receiver. Synchronizer parameters were given in Section 5.8. For all channels, detection performance shows good values, since 1 dB SNR in 2160 MHz bandwidth is sufficient to achieve reliable detection *without* false alarms, which have never been observed. For CM2.2, the influence of phase noise is investigated. Compared to older results shown by the author in [M. 07], a residual FER of 2% could be eliminated with a redesign of the PHY, preamble and synchronization parameters (table 8). Channel estimation suffers from phase noise at higher values beyond 20 dB, as shown in Figure 59. Carrier frequency offset performance shows the expected dependency on phase noise.

Timing performance is acceptable for all channel models. In this respect, CM3.1 is also shown, having the largest delay spread. A short coming of the scheme is the channel estimation performance for CM3.2. For chis channel, the assumption of a short channel delay spread is no longer valid, and the smoothing filter itself produces interference by filtering out delay components. In this case, the filter should be switched off. The timing performance of the coarse synchronizer is also shown in Figure 60. This has been done to exclude the possibility that channel estimation performance is affected by bad coarse timing performance.

For CM3.1, which results from an earlier simulation, timing performance is acceptable for a minimum SNR of 2 dB.

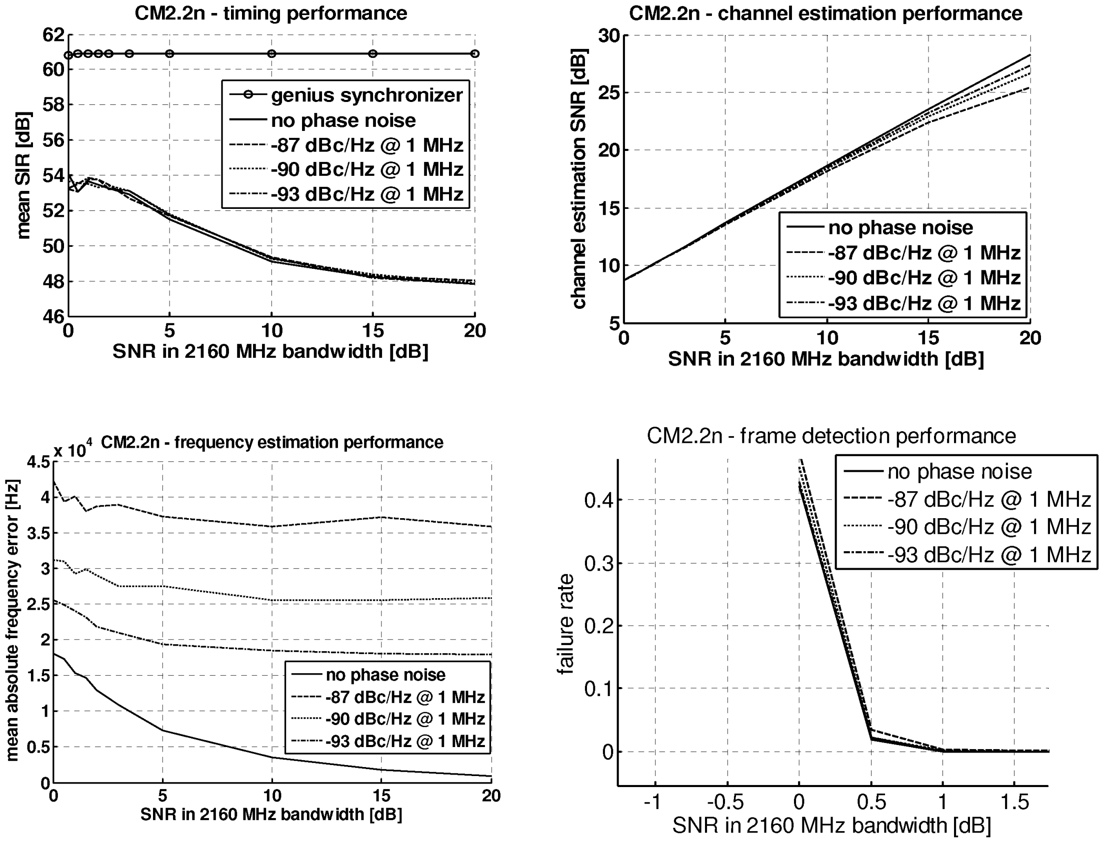


Figure 59: Synchronization performance for CM2.2

5.12.3 Performance of wideband PHY in static channel

The performance of the wideband mode is has been investigated using all receiver algorithms for the residential scenario (CM1.2 for the LOS and CM2.2 for the NLOS channel). Results are shown in Figure 62. Simulations were done with 2048 byte packets for four different cases:

1. convolutional code performance without outer RS encoding for CM1.2
2. convolutional code performance with outer RS encoding for CM1.2
3. convolutional code performance with outer RS encoding for CM2.2
4. convolutional code performance including softbit quantization for CM2.2

Due to very high power fluctuations, every channel response of the NLOS channel CM2.2 was individually normalized. Hence, the performance curve is given for the experienced instantaneous SNR at the receiver and does not account for small scale fading.

BPSK-1/2 has been excluded from this simulation, because RS-encoding leads a performance, which is rather limited by synchronization than by the code performance. Summarizing the curves, we can make some conclusions. RS outer encoding gives a high performance gain, which is most obvious for

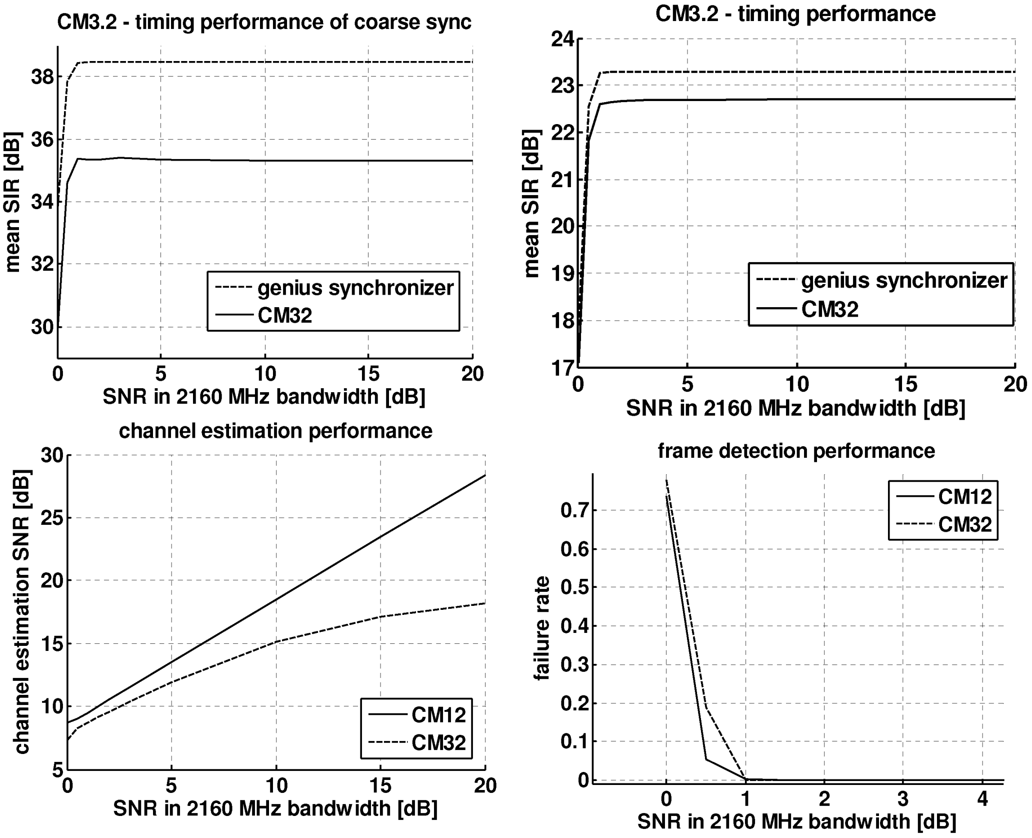


Figure 60: Synchronization performance for CM3.2

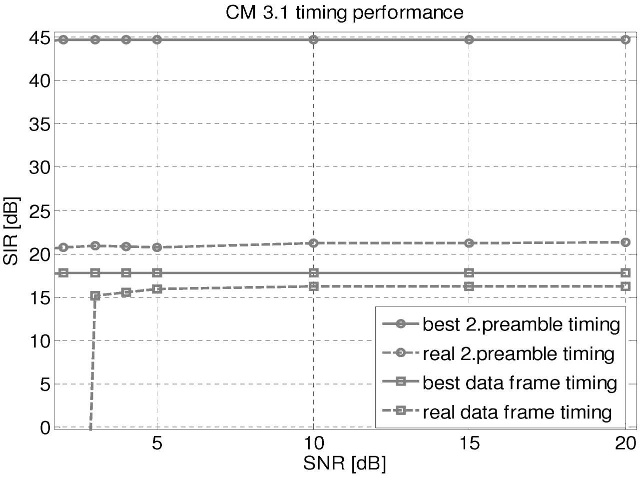


Figure 61: Timing synchronization performance for CM3.1

the punctured modes using 16-QAM. However, the punctured modes tend to fail for the high-frequency selective NLOS channel. The same phenomena was already observed for the narrowband PHY in the HHI channel. 5 soft bits have been used for the bit metrics. Although quantization thresholds have been optimized, a residual degradation for the NLOS channel cannot be avoided.

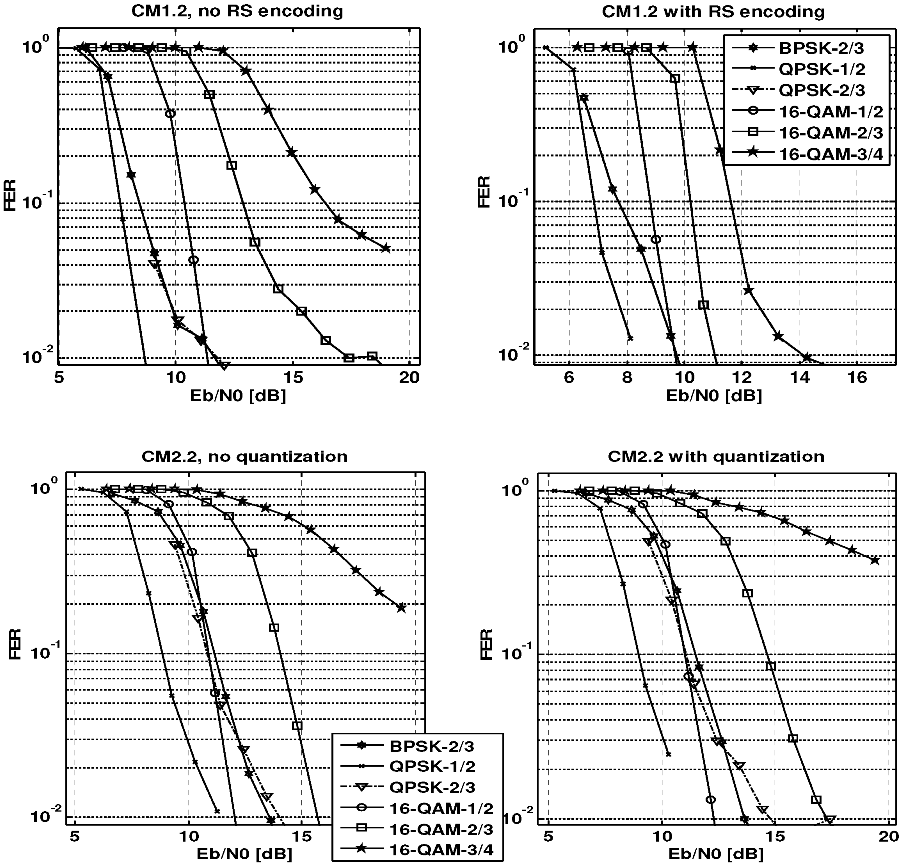


Figure 62: Wideband performance for all modes except BPSK-1/2

5.12.4 Performance of wideband PHY in residential time-variant NLOS channel

Until know we have investigated system performance for a static channel scenario. In general, the associated channel responses are bandlimited by the baseband filter and follow Equation (237) in Appendix B.3. As shown in Appendix B.4, it is possible to derive a time-variant deterministic channel model if angle-of-arrival information is available for the receiver side. Since this is the case for the TG3c channels, we can study the performance of the wideband system for a moving receiver. This has been done for the residential NLOS channel CM2.2, where the transmitter uses a 60 degree antenna beam width. We assume an omni-directional antenna for the receiver to gain the largest amount of multipath components. The channel was again normalized for each channel response, but only based on the snapshot at $t = 0$. However, it was found that the received power is hardly changing during frame reception.

We assume a speed of 10 meters per second or 36 km/hour, which is higher than ever required for office or domestic scenarios. 16-QAM modulation with a code rate of r=1/2 is chosen with RS outer encoding. A phase noise value of -93 dBc/Hz at 1 MHz for transmitter and receiver VCO is expected. The PA is driven with an input power backoff (IBO) of 8 dB. The AGC is enabled, and bit metrics are represented with 5 bits, as in the implementation. The receiver is always moved in the direction of the transmitter.

In order to obtain long frames, we use *MAC frame aggregation*. This method reduces the preamble overhead. The defined PHY layer allows transmission of up to 8 consecutive MAC frames by storing frame-specific data (mainly the data length) in the signal field. We assume that the transmitter can synthesize the next frame such that only the erroneous MAC frames have to be retransmitted. In addition, a reliable back channel shall exist in a full-duplex link. Then we can distinguish between a MAC frame error rate $\mathrm{FER}_{\mathrm{mac}}$ and a PHY frame error rate $\mathrm{FER}_{\mathrm{phy}}$. For the latter, all frames are counted where at least one frame is lost, i.e. one error is made in the receiver. As usual, the frame length is chosen to 2048 bytes. Data is transmitted over 15 convolutional streams. We consider four cases, summarized in table 11. Performance is given in Figure 63 as the achieved PHY and MAC frame error rate for the Eb/N0 ratio, including preamble overhead etc. The frame length is around 100 OFDM symbols. Each channel update takes effect only after a delay of 14 symbols. In this way, we account for the latency of the processor implementation.

Table 11: Link conditions and baseband mode

(1)	static channel CM2.2, no midambles transmitted
(2)	receiver speed = 10 m/s, no midambles transmitted
(3)	receiver speed = 10 m/s, midamble transmitted after every 35 data symbols
(4)	receiver speed = 10 m/s, midamble transmitted after every 18 data symbols

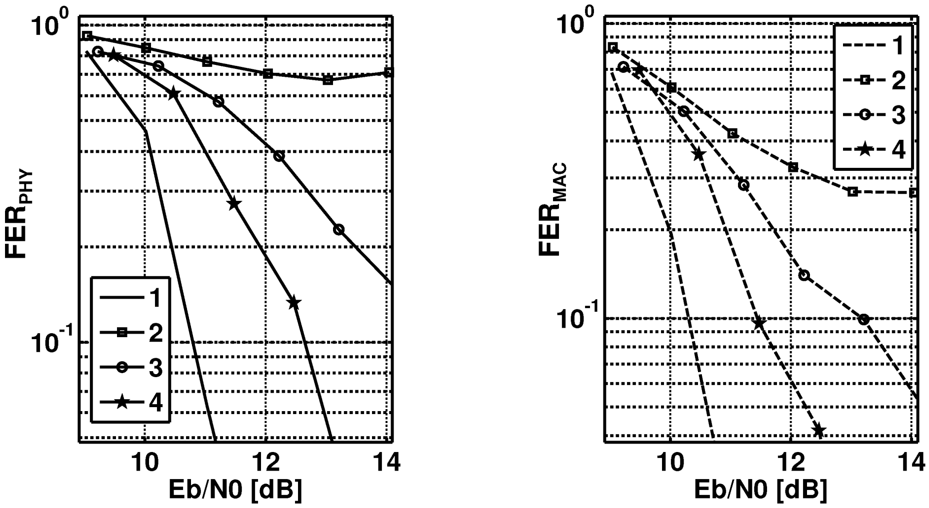


Figure 63: FER performance

Since midamble transmission causes a decrease in data rate, it is also instructive to investigate the netto throughput. Therefore, Figure 64 shows the achieved data rates depending on the SNR in 2160 MHz bandwidth for the different configurations. With negligible inter-frame guard times, this throughput or *goodput* is given as the number of message bits, divided by the frame duration and multiplied with $(1 - \mathrm{FER}_{\mathrm{mac}})$.

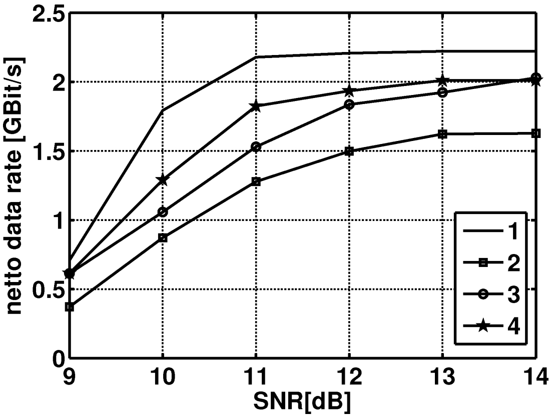


Figure 64: Achieved data rates

It is interesting to see that the system does not completely fail without midamble transmission. Only the netto rate is reduced from 2.2 to 1.6 GBit/s at high SNR. Transmitting midambles after every 18 symbols causes a rate reduction of 10%, but this configuration works best compared beside transmission for the static scenario. Finally, it is noted that in case of a Dirac channel, the tracking mechanism would take care of the time variance.

6 Baseband processor implementation

In this chapter, the implementation of the WIGWAM 60 GHz demonstrator is presented, which uses the narrowband PHY. PHY parameters are given in Appendix G. The core implementation processes an FFT bandwidth[1] of 400 MHz. Blocks designed by the author are described in detail. In addition, redesigned components are discussed for a demonstrator utilizing the wideband PHY. By the time when this work was written, the baseband processor for the wideband mode was still under development. The presentation is focused on the receiver, where most of the hardware effort is usually spent and where more involved processing algorithms are needed. Details regarding the hardware platform can be found in Appendix I. Elementary hardware blocks are introduced in Appendix H. In the following text, some formulas from previous chapters are reproduced at some points for better readability.

6.1 Strategy for FPGA-based processor designs

Some practical guidelines have grown from the initial experience of VHDL-based code design for the envisaged FPGAs. The target had been to minimize the required resources.

- Previously developed ASIC-based designs for FFT and Viterbi decoder turned out to consume very large amounts of resources, because they were based on flipflops and logic only. It was found that RAM-memory-based design blocks require far less FPGA resources due to the large number of available memory blocks.

- It has been found that a frequency of 80-100 MHz is a suitable, convenient clock frequency for synchronous designs on the Virtex-2. A clock frequency in this range can be efficiently handled by the synthesis tool even with a chip utilization beyond 90%. Considerably lower clock speeds might sacrifice computational power in many cases whereas higher clock speeds may require excessive pipelining. In addition, area consumption may likely rise due to automatic gate duplication performed by the synthesis tool in order to meet timing requirements. Synthesis optimization cycles can largely extend for rising frequency. For the same reasons it was found that a clock speed of 150-160 MHz is a practical value on the Virtex-4.

A clock frequency of 100 MHz has been chosen for the baseband processor, which is a quarter of the FFT sampling rate.

6.2 Transmitter

The transmitter baseband architecture is shown in Figure 65. A detailed description of the hardware was published in [M.K08]. Here, we consider only the general functionality. Source data arrives as multiples of bytes. However, the processing in the transmitter is bit-oriented. The first step consists in scrambling of binary data in order to avoid long series of zeros or ones and to avoid retransmission of the same waveform. Long series of zeros or ones could cause very high power peaks in an OFDM symbol and

---

[1] This is the bandwidth processed by the IFFT and FFT. For the narrowband PHY, 333 MHz are really used for the signal, whereas the wideband signal occupies around 1757 MHz bandwidth.

have to be avoided. The scrambler seed is chosen randomly to produce a different scrambling sequence for every frame. Hence, a retransmitted frame will not have the same waveform. If a very high power peak occurred the first time, it will at least not be reproduced the same way during retransmission. The next block would be the byte-processing Reed-Solomon encoder, which has not been integrated in the narrowband implementation, but is an optional part of the MATLAB model.

Depending on the data mode, up to 12 convolutional encoders are involved in the transmission. Such an encoder is shown in Figure 91 in Appendix D.2. The data distribution logic splits source data in blocks and feeds the encoders. Each block contains data to fill an OFDM symbol. The streaming arrangement has been discussed for the wideband mode in Section 5.4. For the narrowband mode, streaming is done in a similar way. From this point, processing is organized for data bursts. The encoded data streams are processed by up to 6 parallel interleavers. Therefore, glue logic is needed to collect and redistribute encoded data. The block size of the interleavers is equal to the data amount in one OFDM symbol, which varies depending on the mode.

Prior to inverse FFT operation, data must to be assigned to the right subcarrier positions of a 256-wide vector of complex numbers, the subcarrier symbols. The interleaved data substreams go to the used data subcarriers. In addition, a pseudo-random sequence feeds the pilot subcarrier positions. The subsequent mapper creates BPSK-, QPSK- or M-ary QAM-symbols for the used data subcarriers and BPSK-symbols for the pilots. Transmission starts with the preamble, followed by the IFFT-transformed signal field (SF) and data frame. Prefix insertion happens inside the IFFT. A separate block is used to create control data mapped into the signal field. A global controller takes care of the BPSK-modulated signal field data, which is processed prior to source data.

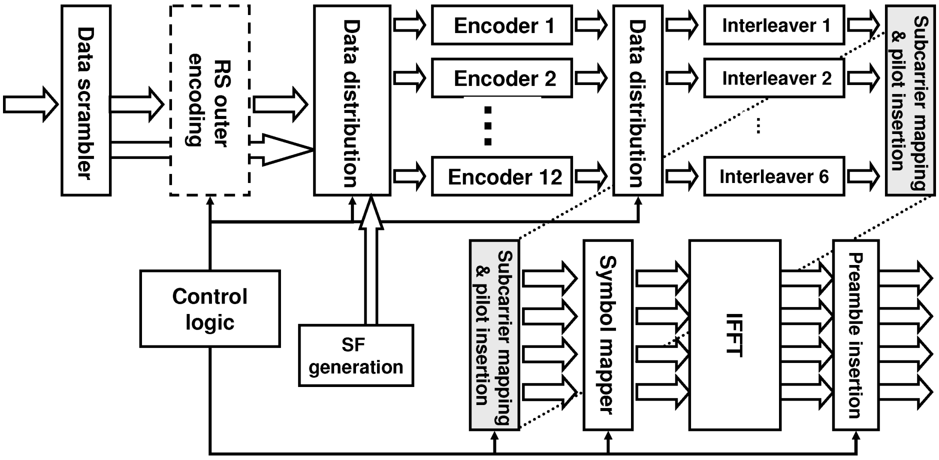


Figure 65: Transmitter baseband architecture

6.3 Receiver overview

The architecture of the digital receiver baseband is shown in Figure 66. This receiver has been published in parts by the author in [Max09b] and [M.K08]. In this and the following sections, a more detailed view is given. We start with an overview, followed by more detailed descriptions of the individual blocks.

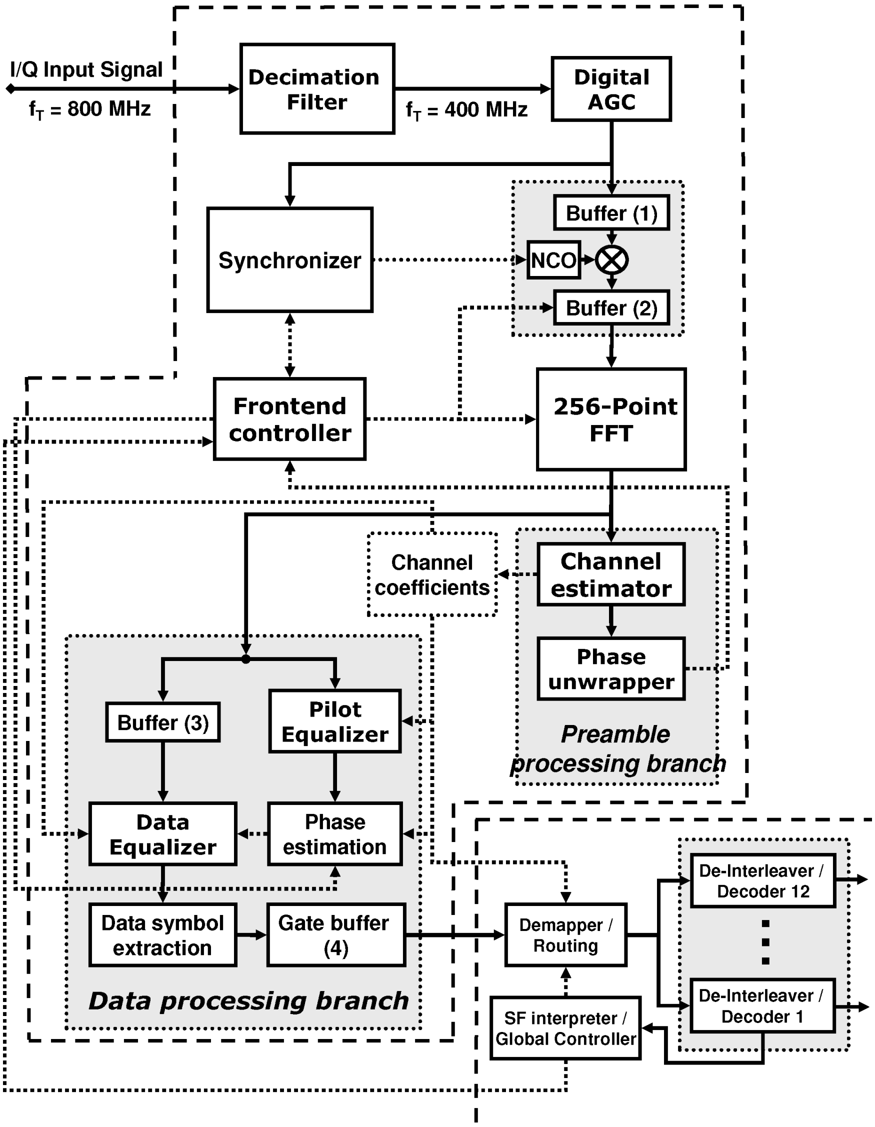


Figure 66: Receiver baseband processor architecture

This receiver has been the first IHP baseband implementation for 60 GHz, which took place during the WIGWAM project. Certain features were not yet implemented: channel re-estimation, channel smoothing and improved timing estimation. In comparison with the later estimation scheme where a smoothing filter is employed, the estimator utilizes a longer preamble of four instead of two preamble symbols to achieve about the same accuracy. The timing estimator requires the prediction filter in order to attain acceptable performance, as previously described.

The architecture follows the principle of a functional separation between the *inner receiver* and an *outer receiver*, as suggested in [Hei98]. The inner receiver performs synchronization, channel estimation, demodulation and equalization. It delivers subcarrier symbols to the outer receiver, which in turn accomplishes demapping, deinterleaving and channel decoding. The inner receiver will be denoted as the *digital front end*. To process a system bandwidth of 400 MSPS, four consecutive signal samples or subcarrier symbols are processed in parallel in most of the stages of the front end. Full pipelining is required to achieve the required throughput. The processing in the blocks after and including the FFT is burst-oriented.

Implementation

The high-speed FPGA development board designed for the demonstrator fond in the Appendix has been used. The front end design fits in a Virtex-2-PRO-100 FPGA and consumes 60% of the slices. The design needs 26560 slices, 36555 flipflops, 40962 4-input lookup tables, 112 block RAMs, and 143 multipliers.

Input block

The analog baseband I/Q signal is two-times oversampled at a rate of 800 MSPS (Figure 67). This oversampling alleviates the requirements for the analog anti-aliasing filter. A halfband 47-tap filter attenuates the upper frequency band from 228-400 MHz by more than 50 dB and delivers the down-sampled sample stream at 400 MSPS to the core unit.

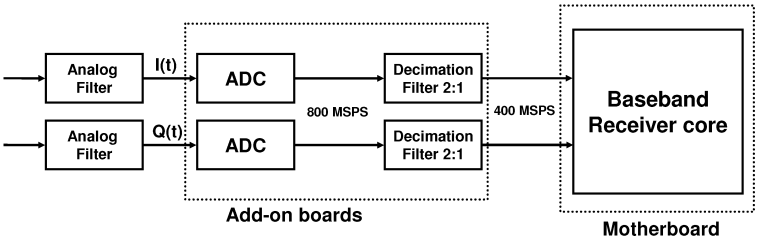


Figure 67: Receiver input stage

In a standard, single-channel digital receiver, A/D converters are driven with a defined signal level in order to avoid clipping for exceeding input amplitudes or excessive quantization noise for very weak input signals. This is accomplished with an automatic gain control (AGC) loop. DSP blocks like the synchronizer also benefit from a constant input signal level, since dynamic range requirements are reduced meaning that signals can be represented with less bitwidth. Unfortunately, an AGC could not be developed in the analog domain due to unavailable functioning radio signal strength indicator (RSSI). Therefore, analog signal level had to be manually controlled in the demonstrator. To establish at least a defined signal level for the DSP blocks, a *digital AGC* has been included (see Figure 66). This block is discussed in Section 6.11.

The *synchronizer* performs frame detection, coarse timing estimation and carrier frequency offset (CFO) estimation. It prompts the AGC to freeze the current gain level as soon as the frame is detected. Otherwise, channel estimation would suffer from amplitude variations of the preamble waveform. Due to processing latency, frequency offset information is only available after the second preamble part has already started. Therefore, a buffer stage is included in parallel to the synchronizer. Right after setting the frequency of a *numerical controlled oscillator (NCO)*, the system starts to continuously read out the first FIFO buffer, correct the frequency shift by complex multiplication with a rotating phasor and write the result in a second FIFO buffer. When setting the offset frequency, the synchronizer also announces a new successfully detected frame and provides the start index for the first FFT.

Front end Controller and FFT

The ’intelligence’ of the receiver is shared mainly between two controllers, the *front end controller* and the *signal field interpreter*. The front end controller manages global flow control of the digital front end. We may term the complete FFT transformation of a data burst of 256 samples as an *FFT operation*. The front end controller consists of a state machine for sequencing and initiation of FFT operations at defined times. Transfers of data bursts into the FFT are managed by a subcontroller. An FFT operation can be initiated as soon as the first sample of the related sample sequence is available in the second buffer. The front end controller also determines whether FFT output data is sent to the *preamble processing branch* or the *data processing branch*.

The 256-point FFT has been implemented according to the radix-4 algorithm. Four 64-point sub-FFTs are combined in a radix-4 butterfly to make up a 256-point FFT with a throughput of four-times the sampling rate. The FFT algorithm is recalled in Appendix F and the implementation is discussed in Section 6.8.

Preamble and data processing processing branch

The preamble processing branch (PPB) carries out channel estimation and fine frame synchronization using the four symbols of the second preamble part. The inverse channel coefficients are stored in polar coordinates for the data processing branch. In addition, the subcarrier power coefficients are calculated for the symbol demapper. The choice of polar coordinates enables the receiver to easily change the time reference without big effort, as will be explained.

The equalizer in the data processing branch (DPB) uses the *zero-forcing* equalization scheme, where each subcarrier coefficient is compensated by channel inversion. Noise amplification is taken into account by weighting of the bit metrics. Since inverse channel coefficients are supplied in polar coordinates, the data equalizer can do equalization and phase error correction in one step. For this purpose, the continuous pilot subcarriers need to be processed first in the *pilot equalizer* and *phase estimator* to provide estimates for the common phase error and the timing error for the current OFDM symbol. Meanwhile, data subcarriers are stored in buffer 3. Only after data equalization, the data symbols are actually extracted from the subcarrier block, dropping outer guard subcarriers and DC zero subcarriers. A *gate buffer* is used to prevent discarded data to reach the outer receiver. Data needs to be discarded in some cases as explained when the scheduling scheme is considered. The FFT and each block of the DPB can process data without wait cycles. Since the received OFDM symbols feature a cyclic prefix and this prefix is dropped, the maximum throughput of the digital front end exceeds the input rate by a factor of $\alpha_R = (T_g + T_{FFT})/T_{FFT} = 1.2$.

Data path processor (outer receiver)

The outer receiver contains a bank of 12 parallel decoding blocks. Each block comprises a de-interleaver and a Viterbi decoder and processes up to 100 MBit/s. The number of involved code streams during a frame reception is defined by the transmitter and stored as a parameter in the signal field. A minimum number of decoders is required for each data mode. The receiver is capable of a higher throughput for all data modes, since the processable peak rate is 1200 MBit/s. The front end exceeds the minimum throughput by a factor of $\alpha_R = 1.2$. The datapath processor may utilize more decoders than required. This can be exploited to gradually reduce the initially accumulated latency caused by synchronization and preamble processing.

Scheduling

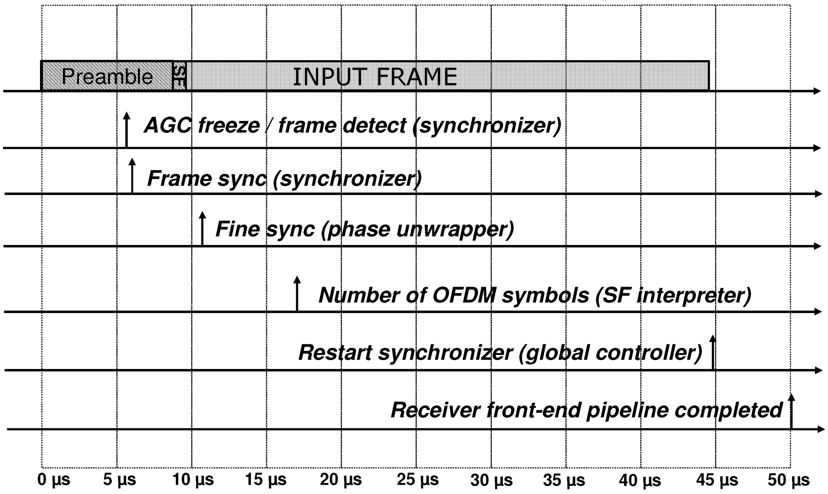


Figure 68: Receiver scheduling with real timings

The front end controller has to wait until finer timing has been estimated by the phase unwrapper. With re-adjusted FFT positioning, the receiver starts the processing of accumulated waveform data. The first data symbol in the frame constitutes the signal field, which defines the transmission mode parameters. These are available when the signal field has passed the complete chain consisting of the data processing branch in the front end and the demapper, de-interleaver, decoder and signal field interpreter. Given a transmission without immediate acknowledgment, some interframe guard time is necessary for the receiver to be able to process the next transmitted frame. Two provisions keep this guard time short.

- Firstly, instead of waiting for reply from the interpreter, the receiver immediately starts to process the subsequent accumulated data symbols regardless of the actual frame length. An intelligent buffer stage right before the front end output acts as a gate for the data flow. It lets the signal field through but keeps the next OFDM symbols in the queue until frame length information is available. For short frames or in case of an erroneous signal field, some stored symbols will be discarded.

- Secondly, the receiver is able to handle two consecutive frames simultaneously. As soon as the last OFDM symbol is sent to the data processing branch, the front end controller reactivates AGC and synchronizer. While the rest of the receiver may still process the older frame, synchronization and channel estimation can already take place for the next frame. Data confusion is avoided with paging. Every OFDM symbol carries a frame label, which is used to allocate the dedicated channel data.

A diagram of the receiver scheduling with real timings is shown in Figure 68.

6.4 Coarse synchronization and CFO estimation block

The synchronization algorithm has been described in Section 5.8 in a pure mathematical fashion, and we now consider its implementation. For an overview, a block diagram of the architecture is shown in Figure 69. This structure is utilized in both the narrowband and the wideband system. The first block is used to compute the sliding autocorrelation given in Equation (120). The *antiphase detection block* monitors the output of the autocorrelator in order to detect samples for which the *antiphase condition* is met, as described by Equations (125)-(128). Adjacent detected samples are identified as a whole region by the *cluster processing block*, given by Equations (129)-(130). The middle point of each region is called a *peak point*, and the position of each peak point is send to the last block, which contains the control logic of the synchronizer as a finite state machine (FSM). In Section 5.8 it was shown that two such peaks must be found when the preamble is received, and that they must occur with a pre-defined time separation. Therefore, the FSM measures the time between the occurrence of the first and a possible second peak and discards the frame if the distance is outside the tolerance range (Equation (131)) or if the second peak is missing, causing a time-out. After the frame has been detected, the long ACF is evaluated at two positions for the estimation of the carrier frequency offset. In the following, each block is described in more detail.

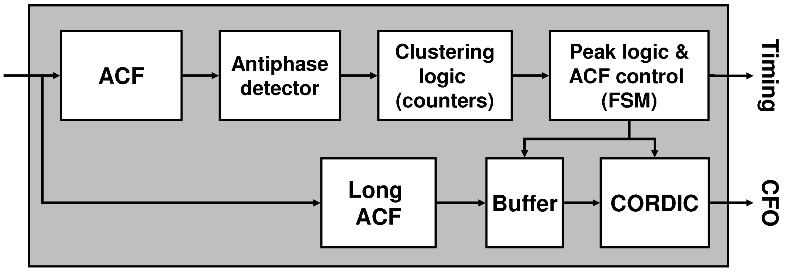


Figure 69: Synchronizer block diagram

6.4.1 Autocorrelator

The sliding autocorrelator has to compute the numerator and denominator of Equation (120). This equation is reprinted together with (119) for convenience.

$$A_x(n, N_\Delta, N_I) = \frac{\sum_{k=0}^{N_I-1} \overline{x(n-k-N_\Delta)}x(n-k)}{\max\{P_x(n, N_I), P_x(n-N_\Delta, N_I)\}} \tag{175}$$

$$P_x(n, N_I) = \sum_{k=0}^{N_I-1} \overline{x(n-k)}x(n-k) \tag{176}$$

The autocorrelation block mainly needs to compute cross products $\overline{x(n-N_\Delta)}x(n)$, instantaneous power products $\overline{x(n)}x(n)$ and moving averages of these products. Note that the input sample stream arrives on L parallel ports with L=4 for the narrowband and L=8 for the wideband system, since the clock frequency is only a fraction of the sampling rate. This leads to higher implementation effort resulting in larger chip area and potential timing issues. In general, a moving average over d numbers for some input $g(n)$, producing an output signal $y(n)$, is given as

$$y(n) = y(n-1) + g(n) - g(n-d) \tag{177}$$

The output $y(n)$ works as an accumulation register. Unrolling this equation for L input ports, where L adjacent sample phases arrive simultaneously, leads to the following equation set.

$$\begin{aligned} y(Ln) &= y(Ln-1) + g(Ln) - g(Ln-d) \\ y(Ln-1) &= y(Ln-2) + g(Ln-1) - g(Ln-d-1) \\ &\cdots \\ y(Ln-L+1) &= y(Ln-L) + g(Ln-L+1) - g(Ln-d-L+1) \end{aligned} \tag{178}$$

To arrive at regular structures, we assume that d is dividable by L. According to (178), it would be required to delay L different phases simultaneously. In FPGAs, delays can be implemented with embedded static RAM blocks (Appendix H.4), which are available in large amounts.

We assume that the L output values are registered. If the equation set (178) is implemented in a straightforward way, only one output register is reused for the next iteration, $y(Ln)$. Therefore, this register figurates as a accumulator as in the single-port case. The longest calculation path exists between $y(Ln-L)$ and $y(Ln)$ and consists of L additions and subtractions. The subtractions $g(Ln-l)-g(Ln-d-l)$ for l ranging from 0 to $L-1$ can be pre-computed in a previous pipeline stage. This reduces the number of operations to L additions. For the wideband system, L=8 additions are still critical in terms of timing, because the Virtex-4 device has to run at a relatively high frequency of 270 MHz.

The implementation effort and timing constraints can be reduced if the ACF is not evaluated for every new sample position. In fact, it was found by simulation that the performance loss is negligible for both the narrowband and wideband case, if the ACF is evaluated only for every L-th sample. We may select phase $y(Ln)$ and omit all other phases $y(Ln-1)$ to $y(Ln-L+1)$. Combining the equations in (178), we can write for the selected phase

$$\begin{aligned} s(n) &:= \sum_{k=0}^{k=L-1} g(Ln-k) \\ y(Ln) &= y(Ln-L) + s(n) - s(n-d/L) \end{aligned} \tag{179}$$

Applying this simplification on the moving average calculations, we arrive at a structure for the autocorrelator shown in Figure 70 for the narrowband case with L=4, $D \equiv N_\Delta$ and $M \equiv N_I/L$. The memory requirements are reduced by a factor of L. The outputs $Y(n)$ and $P_{\max}(n)$ are the numerator and denominator of the normalized ACF in Equation (175). The downsampling also greatly simplifies processing in the subsequent stages, which are discussed next.

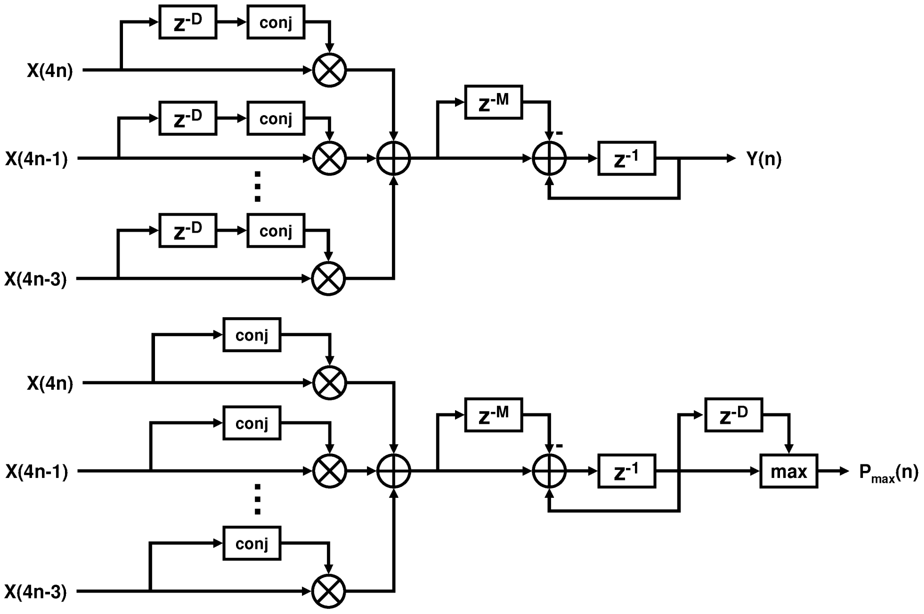


Figure 70: Autocorrelator for 4 input ports and 1:4 rate reduction

6.4.2 Antiphase detector

The antiphase detector is depicted in Figure 71. Its function is to detect samples near the area, where the preamble is flipped in sign, as shown in Figure 48. We adopt a simpler notation than in Section 5.8. During the phase transition, the complex ACF output $Y(n)$ will change its angle by 180 degrees. Since the angle of $Y(n)$ depends on the frequency offset and is unknown at this stage, the sample detector relates the ACF output to a delayed version $Y(n-D_0)$. More precisely, it tests if $Y(n)$ and $Y(n-D_0)$ both exceed a given amplitude threshold and if they are approximately 180 degree out-of-phase. For this purpose, $Y(n-D_0)$ is rotated to align with the real axis, and $Y(n)$ is flipped in sign and rotated by the same angle.

$$\begin{aligned} Y_1(n) &:= Y(n-D_0)\exp(-j\angle\{Y(n-D_0)\}) = |Y(n-D_0)| \qquad (180)\\ Y_2(n) &:= -Y(n)\exp(-j\angle\{Y(n-D_0)\}) \end{aligned}$$

This transformation can be conveniently accomplished with a pipelined CORDIC processor operating in dual-mode. This block is described in Appendix H.1. The detection conditions are implemented in the subsequent *detection logic* block. Since $Y(n)$ is the numerator of the normalized ACF, conditions (126)-(128) transform into

$$\begin{aligned} Y_1(n) &> \alpha_1 \cdot P_{\max}(n-D_0) \qquad (181)\\ \mathrm{Re}\{Y_2(n)\} &> \alpha_2 \cdot P_{\max}(n) \qquad (182)\\ \alpha_3 \cdot \mathrm{Re}\{Y_2(n)\} &> |\mathrm{Im}\{Y_2(n)\}| \qquad (183)\\ &\Rightarrow n \in \mathcal{M} \end{aligned}$$

where, again, $\mathcal{M}$ denotes the set of detected samples. Since $Y_1(n)$ aligns with the positive x-axis after CORDIC operation, no calculation of the magnitude is necessary. The binary output $f(n)$ marks detected samples. It has to be mentioned that the output of each block is accompanied with a time index. In this way, the last block (FSM) knows which samples in the stream are detected.

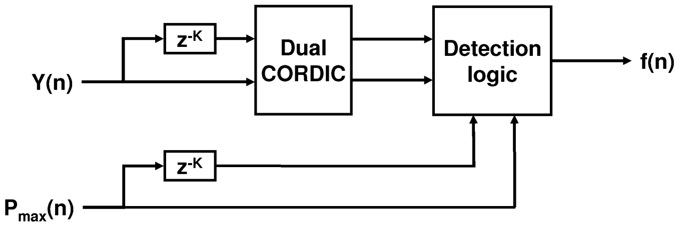


Figure 71: Antiphase detector

6.4.3 Clustering logic, main controller and long autocorrelator

The clustering block is made up of simple logic. The idle state is defined by a counter being in zero state. The first time when a detected sample arrives ($f(n) = 1$), the time index of that sample is stored and the counter is set to d_1, the maximum distance between two detected samples regarded to belong to the same region. When the counter is positive, the block is in active state. In this state, the counter is decremented by one if there is no active input sample for the current cycle, or reset to d_1 if there is. In addition, the index of the last active sample is always remembered. As soon as the counter reaches the value one and there is no active sample, the end of the region is reached and the peak point can be computed as the average of the stored first and last index of the region. This value is send to the FSM of the synchronizer. The behavior is illustrated in Figure 72.

As earlier described, the synchronizer controller waits for two peaks to arrive and tests the time difference between them. If a frame is successfully detected, coarse frame timing is obtained and the first

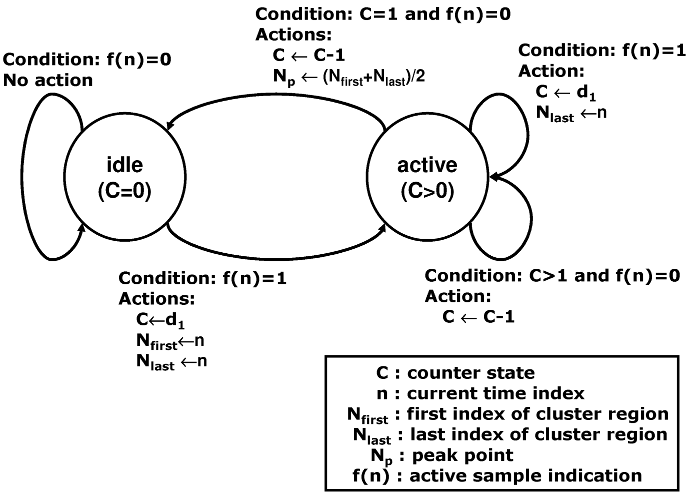


Figure 72: Clustering Logic

peak is taken as a time reference. The next step consists in the carrier frequency offset estimation. The long autocorrelator is evaluated before and after the sign flip of the preamble (Section 5.8, Figure 48). For this purpose, there is a buffer stage inserted after the correlator to compensate for the processing delay of the frame detector. A CORDIC instance is used to calculate the angle of the complex CFO output for the two time indices. This angle is proportional to the carrier frequency offset. After the two estimations are averaged, the estimated CFO value is used to tune the numerically controlled oscillator (shown in Figure 66) for CFO compensation. Finally, timing information is send to the main controller of the receiver, and FFT-based channel estimation can be started.

6.5 Channel estimator and post-FFT timing estimator

After synchronization, four long preamble B-symbols are transformed into frequency domain by FFT operation and send to the *channel estimator*. These symbols are averaged to give 6 dB estimation gain. As discussed in Section 5.6, the B-symbols are generated from some pseudo random BPSK reference sequence $b(k) \in \{-1, 1\}$ in frequency domain. For OFDM, each subcarrier symbol appears multiplied with the channel transfer function taken at the frequency of the subcarrier (see Equation (263) in Appendix C). To estimate the channel coefficients, the receiver only needs to flip the sign of those subcarrier symbols, where $b(k) = -1$. This is done in the reference rotations block. Once the data frame is processed, the receiver does not continuously improve the channel estimate using interpolation or decision-feedback estimation. It can merely perform a completely new channel update based on a midamble. Note that only *derived* values from the estimated channel coefficients $\hat{H}_l$ and not the coefficients themselves are required for equalization. This allows the receiver to calculate polar representations $\hat{A}_l \exp(j\hat{\phi}_l) = \hat{H}_l$ of the coefficients and to store the inverse channel coefficients in dual-port memory blocks as $1/\hat{H}_l = (1/\hat{A}_l) \exp(-j\hat{\phi}_l)$. This is done with a pipelined 4-port CORDIC stage and a divider stage (Appendix H). In addition, subcarrier power gain values are estimated as $\hat{P}_l \approx \hat{A}_l^2$, which are required in the demapper and phase estimation block. Storage in polar coordinates has the advantage that the time base for the coefficients can easily be changed.

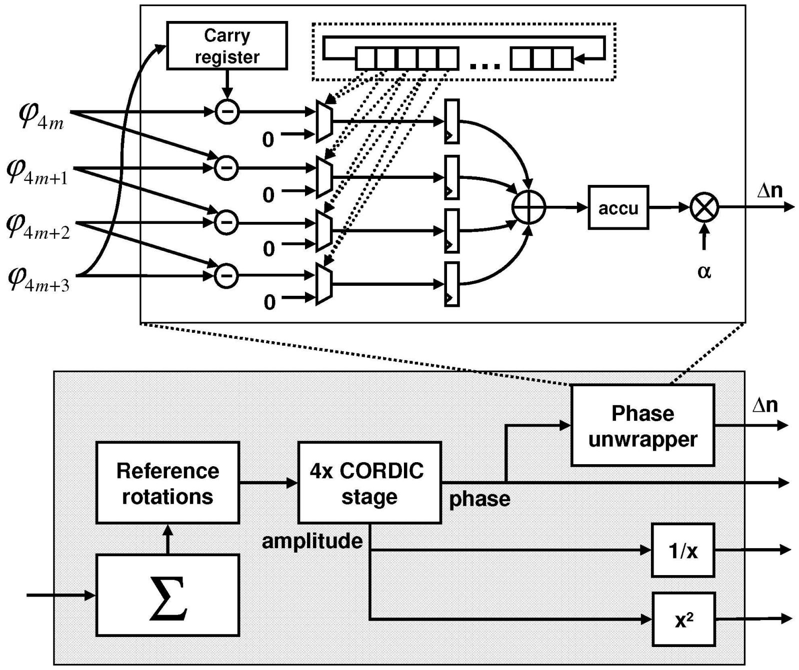


Figure 73: Channel estimator and phase unwrapper

Channel estimation and fine time synchronization were discussed in Sections 5.9 and 5.10. The block diagram for the corresponding hardware blocks is depicted in Figure 73. The timing synchronizer is essentially a *phase unwrapper*. We recall the main equation for convenience. Using the time shifting property of the FFT ([Joh96]), the current FFT window location with respect to a time reference near the *mean excess delay* is proportional to the average phase increase in frequency domain and can be estimated via

$$\triangle n \approx \left(\sum_{l \in \mathcal{M}} \left[\hat{\phi}_{l+1} - \hat{\phi}_l \right]_{2\pi} \right) \cdot \alpha / \pi \qquad \alpha = N_{FFT}/(2N_{pair}) \tag{184}$$

$[\ldots]_{2\pi}$ shall denote a complementary modulo-function defined as $[\phi]_{2\pi} = [(x+\pi) \mod (2\pi)] - \pi$. The sum runs over the set $\mathcal{M}$ of those $N_{\text{pair}} = 206$ indices l, for which subcarrier l and subcarrier $l+1$ are both used for transmission (as active pilots or data subcarriers). This wrap around arithmetic is required to avoid phase ambiguity. The receiver readjusts frame timing by choosing a fixed offset in advance of the estimated time reference. This results in some sample offset $\triangle m$ from the new to the old FFT window position. Due to the shifting property, it means that not the inverse of the original estimates $\hat{H}_l$, but the inverse of

$$\hat{G}_l := \hat{H}_l \exp(j2\pi \triangle m \cdot l / N_{FFT}) \tag{185}$$

are required for equalization. Having the channel coefficients stored in polar representation, transformation to the new set of coefficients is trivial and can be done in the equalizer. The receiver only needs to store the time offset $\triangle m$ as a global parameter.

For the phase unwrapper, a 1-bit circular shift register of length $N_{\mathrm{FFT}} = 256$ is embedded to identify active subcarriers. The output is used to switch multiplexers to give either zero output or the difference between each two valid inputs. To realize the wrap-around arithmetic, the phase differences are performed without 1-bit range extension. With four multiplexers, some subtractors / adders and registers and some final multiplication, the additional complexity to perform fine timing synchronization on top of channel estimation is quite low. This scheme requires only one subtraction and two additions *per input sample*. In a typical OFDM system, a cross correlator is used for fine frame synchronization ([M.J07]). If cross-correlation with pattern length N is carried out in full complexity, N multiplications and $N - 1$ additions are required per sample. Complexity can be reduced by quantization of the correlation pattern. Even if the correlation pattern is quantized with one bit, there are still $N - 1$ additions to perform per input sample.

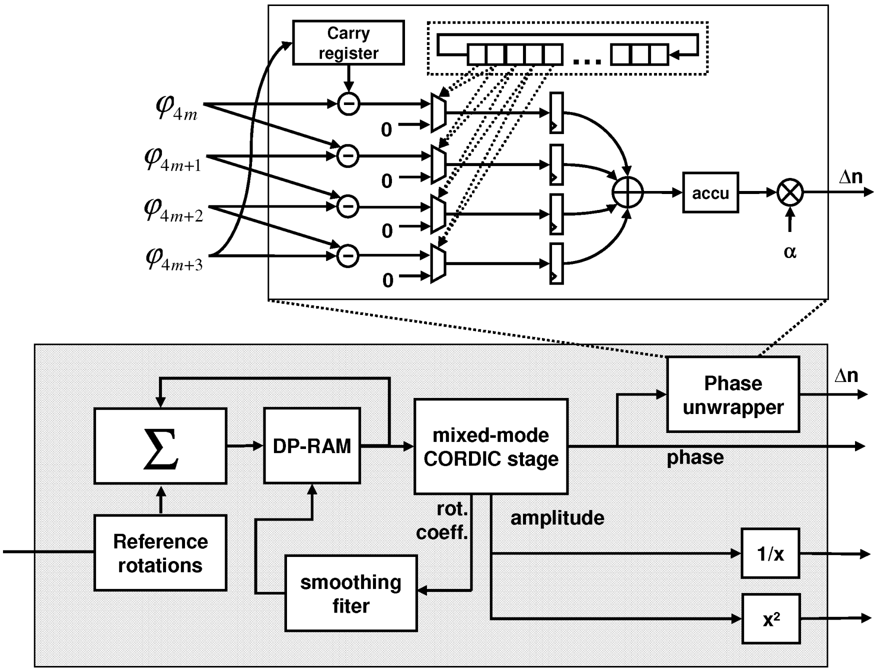


Figure 74: Redesigned scheme with smoothing filter

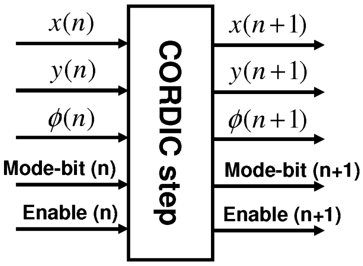


Figure 75: Mixed-mode CORDIC element

The improved channel estimation scheme, which includes a smoothing filter, has been discussed in

Section 5.9.1. The main required steps are given below.

1. An initial channel estimate is performed in frequency domain using FFT transformation, reference rotation and averaging.
2. Phase unwrapping is carried out for fine timing estimation. This unwrapping can already start when channel estimation is still performed and causes little latency.
3. The channel coefficients are rotated to obtain a zero average phase increase, i.e. a shift of the main power of the channel impulse response to the time origin.
4. Filtering of the coefficients in frequency domain is done to reduce noise components far away from the mean excess delay.
5. Channel coefficients are transformed from Cartesian- to polar-coordinate representation, and power coefficients are calculated in parallel.

This scheme can be efficiently realized with a hardware structure as shown in Figure 74. The key element is a mixed-mode CORDIC processor, which is used three times. In Appendix H.1, the well-known CORDIC algorithm is recalled. This algorithm can work in vector- and in rotational mode. Any CORDIC in this receiver is implemented as a cascade of primitive blocks computing one elementary CORDIC step. The mixed-mode elementary block is shown in Figure 75. The direction of a fixed-angle rotation applied on input vector $(x(n), y(n))$ is decided either with the sign of an angle $\phi(n)$, or the sign of the imaginary part $y(n)$. Therefore, with only little additional effort, a CORDIC implementation can be extended to operate in both modes. Input data merely needs to be complemented by a mode bit. In the design, a pipelined mixed-mode CORDIC-stage is used, and the computational steps are given as follows.

1. The initially obtained channel coefficient vector is sequentially stored as complex numbers in dual-port memory.
2. This memory is sequentially read out for phase unwrapping. The CORDIC stage operates in vector mode.
3. After timing has been obtained, the channel vector is read out again for phase rotation and immediately enters the smoothing filter. The output is written back into the dual-port memory. The CORDIC processor is used in rotational mode. An additional block is calculating the phase for each coefficient (not shown).
4. The vector is sequentially read out again to obtain the polar-coordinate representations (A, ϕ) (and $(1/A, -\phi)$ respectively) and power coefficients of the channel coefficients as done in the previous scheme. The CORDIC is used in vector mode.

For the narrowband PHY, four channel coefficients are processed in each active stage in every cycle. For the wideband PHY, the parallelization factor is doubled, but the principle operation is the same. Timing requirements are not exacerbated. Finally, we ask for the allowed latency of the channel estimation block. In the previous case, the global controller can start an FFT operation for the signal field as soon as frame timing is available. Since the FFT itself has latency, the channel-estimation block has time to calculate refined channel coefficients and store them into memory. An analysis shows that the processing is paid only with a latency increase of half of an OFDM symbol duration.

6.6 Pilot machine

The baseband implementation of the narrowband PHY relies on a static environment assuming a time-invariant channel during frame transmission. On the other hand, the receiver needs to cope with fast changing carrier phase and timing offset. The tracking scheme used for the narrowband demonstrator has been discussed in much detail in Section 5.11.1, where the mathematical treatment is given. Here, we only focus on the hardware implementation, depicted in Figure 76. This structure can be used for the original scheme given by Equations (158), (162), (163), (164), (165) and (166), but also for the slightly improved scheme defined by Equations (167), (168), (169) and (170).

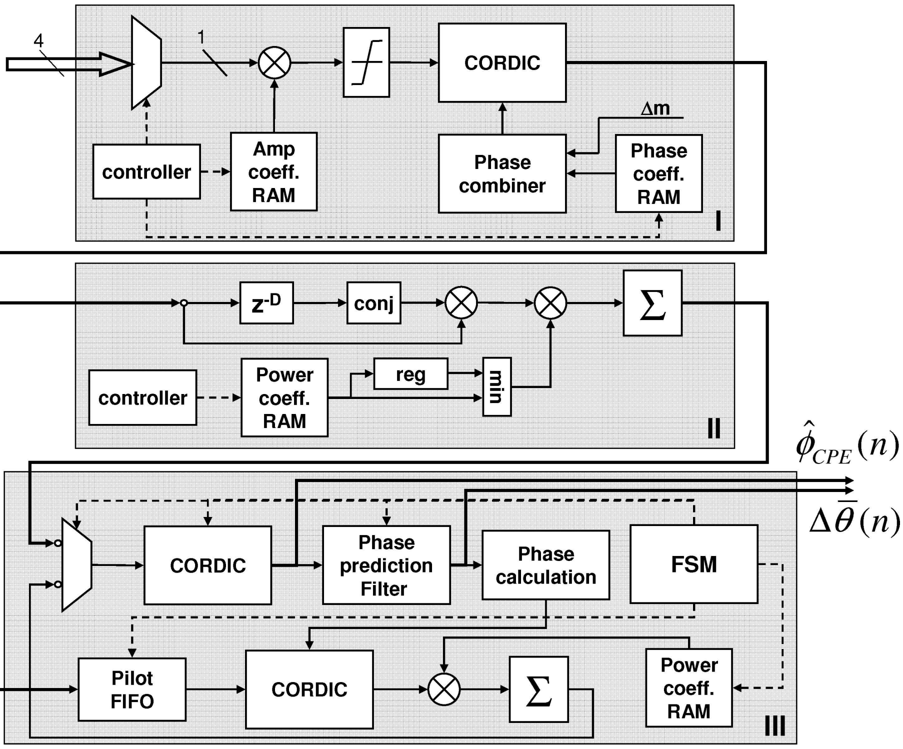


Figure 76: Pilot symbol processor

The pilot processing block will be referred to as the *pilot machine*. In contrast to all hardware blocks, which process data subcarriers, this block does not require parallel busses. With a fixed pilot subcarrier distance of $\Delta_p = 14$, only one pilot can appear each cycle.

The pilot machine is divided into three major blocks. Block I is responsible for pilot equalization and demodulation. The latter operation is done to get rid of the pseudo-random BPSK sequence. Since the inverse amplitude coefficients are stored in memory, pilot equalization can be realized with a multiplier and a CORDIC processor in rotational mode. Sign flipping is managed with the same CORDIC operation by adding a phase shift of 180 degrees. The input of the pilot equalizer is a four-sample wide data port carrying the symbols of data and pilot subcarriers after FFT and shifting operation. A multiplexer and control logic is used to pick the pilot subcarriers at their positions within the data burst of each OFDM symbol. A clipping stage is inserted after amplitude correction in order to limit dynamic range requirements of the system. Under normal operation, pilot subcarriers, which are strongly attenuated by the frequency selective channel and which fall below the noise level, can cause clipping. Using power weighting, such pilot subcarriers have little contribution to the estimation results.

The second block reproduces the calculation of the instantaneous timing estimate based on the current OFDM symbol. This estimate is delivered as a weighted vector sum of complex pilot products, according to Equation (167).

Block III is used for estimation of the final estimated parameters, the averaged phase slope $\Delta\overline{\theta}(n)$ and the common phase error $\widehat{\phi}_{\rm cpe}(n)$[1]. The equalized pilots, which are used in block II, are also required in block III for the CPE estimation and have to be stored in a FIFO. The phase of the instantaneous phase slope estimate is calculated with the upper CORDIC of block III, operating in vector mode. The result is subsequently applied to the prediction filter. After obtaining the improved timing estimate, the pilots stored in the FIFO are phase corrected using the lower CORDIC operating in angular mode. Afterwards, the upper CORDIC is *reused* for the power-weighted sum of the pilots. A 7-state finite-state machine (FSM) is used to control this subsystem. This sequential processing is possible since the upper block I delivers an output parameter with a period of no less than 64 clock cycles.

In the first scheme, adjacent pilots are used as pilot pairs using D=1. For the improved scheme, the predictor is easily redesigned to mimic equations (169) and (170). The delay between pilots in block I is increased to D=8. For the wideband mode, block II must be replicated in order to have a coarse and a fine estimation of the phase slope. The predictor is replaced by a simple unwrapping logic, see Section 5.11.1.

6.7 Data equalizer

Data equalization in this receiver not only performs channel inversion, but also corrects the phase error arising by a residual CFO, phase noise and timing drift. The block itself constitutes a straight-forward implementation which consists of a multiplier stage for amplitude correction and a CORDIC stage for phase correction. A separate block is used to generate the compensation phase for each subcarrier. This phase is a function of the channel coefficient, the estimated phase error and the time basis parameter Δm as previously defined in Section 6.5. A clipping stage is incorporated to limit the output in case of deep fading subcarriers.

---

[1] In section 5.11.1, the parameters are labelled $\Delta\widetilde{\theta}(n)$ and $\widehat{\theta}_0(n)$.

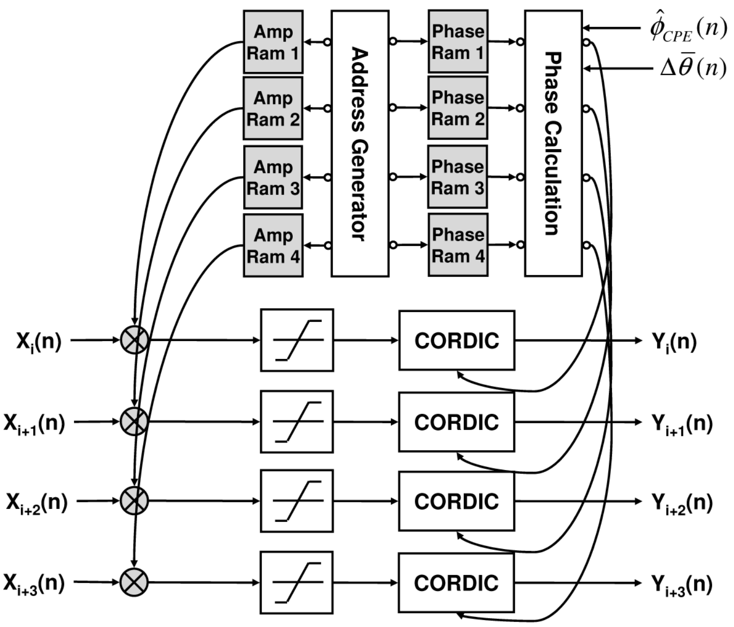


Figure 77: Data equalizer

6.8 Four-port 256-point FFT

In Appendix F, the partitioning algorithm is described, which is the basis for various FFT algorithms. This algorithm can be recursively applied on the sub-DFTs to reduce the overall effort. Very common are radix-r algorithms where the number of coefficients is chosen to $N = r^\nu$ in order to enable $(\nu - 1)$-ary recursive application of Equation (305) ([Joh96]). This means that either the row-DFTs or column DFTs are r-point DFTs. The most common choices are $r = 2$ and $r = 4$, because the r-point DFTs do not require multiplications. More precisely, the twiddle factors for the 2-point and 4-point DFT take the values $W_2^{pl} \in (1, -1)$ and $W_4^{pl} \in (1, j, -1, -j)$. For radix $r = 2$, multiplication in (308) must be performed only for half of the values, since the first row is multiplied with ones. For radix-4, 3 out of four rows must be multiplied. It can be shown that the required number of multiplications is of order $N/2 \log_2(N)$ for radix 2 and $(3N/2)\log_2(N)$ for radix 4.

Since the input is initially stored column-wise and then processed row-wise (Figure 95), the described algorithm is labeled as decimation-in-time. In the WIGWAM-demonstrator, data must arrive and leave the FFT on four parallel ports. Therefore, we can make use of the radix-4 algorithm to split processing of the $N_{\mathrm{FFT}} = 256$-point FFT into four parallel 64-point FFTs $X_l(q)$, $l = 0..3$, and combine the outputs of the FFTs using the so-called radix-4-butterfly. For $L = 4$ and $M = N_{\mathrm{FFT}}/4$ we obtain from (305) and (306):

$$X(p,q) = \sum_{l=0}^{3} \left\{ W_{N_{\mathrm{FFT}}}^{lq} \left[\sum_{m=0}^{N_{\mathrm{FFT}}/4-1} x(l,m) W_{N_{\mathrm{FFT}}/4}^{mq} \right] \right\} \exp(j(\pi/2) \cdot lp) \tag{186}$$

$$= \sum_{l=0}^{3} \left\{ W_{N_{\mathrm{FFT}}}^{lq} \cdot X_l(q) \right\} \exp(j(\pi/2) \cdot lp) \tag{187}$$

In matrix form, this is written as

$$\begin{pmatrix} X(0,q) \\ X(1,q) \\ X(2,q) \\ X(3,q) \end{pmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{pmatrix} X_0(q) \\ W_{N_{\mathrm{FFT}}}^{q} \cdot X_1(q) \\ W_{N_{\mathrm{FFT}}}^{2q} \cdot X_2(q) \\ W_{N_{\mathrm{FFT}}}^{3q} \cdot X_3(q) \end{pmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -j \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & j \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{pmatrix} X_0(q) \\ W_{N_{\mathrm{FFT}}}^{q} \cdot X_1(q) \\ W_{N_{\mathrm{FFT}}}^{2q} \cdot X_2(q) \\ W_{N_{\mathrm{FFT}}}^{3q} \cdot X_3(q) \end{pmatrix} \tag{188}$$

For each column, the four-input radix-4 butterfly performs 3 complex multiplications with twiddle factors $W_{N_{\mathrm{FFT}}}^{lq}$ and 8 complex additions, as seen from the matrix contents in (188). For the 64-point sub-FFTs, a free available radix-4 FFT IP core from XILINX has been used. The structure of the complete module is shown in Figure 78. The implemented FFT has a maximum throughput of 4 samples per clock cycle without required wait cycles. Since all sub-FFTs are identical and receive the input data synchronously, the outputs are also generated at the same time. They are moved out in serial order. The last block of the 256-FFT is used to recover the natural order of the FFT coefficients $X(k)$. This reordering is not as simple as Figure 95 in Appendix F may suggest. In order to save memory and latency, the sub-FFTs are not reordered, since this can be done in one step at the top-unit.

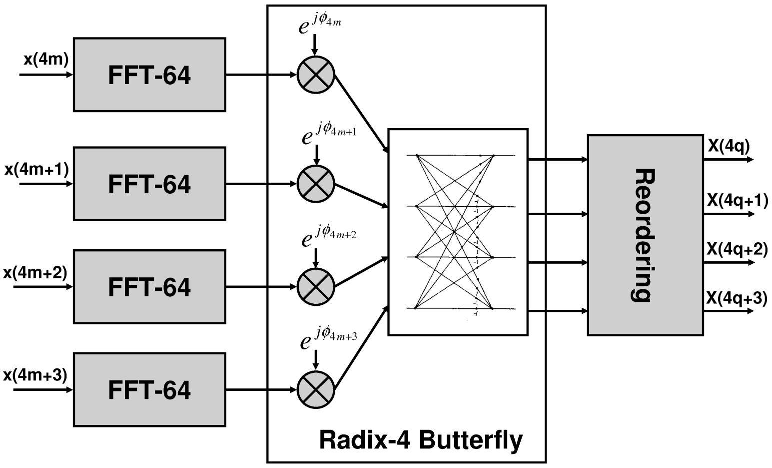


Figure 78: Radix-4 FFT

For the WIGWAM demonstrator, the clock frequency has been kept constant throughout the design. But synthesis shows that the FFT as a highly optimized block can operate at a higher frequency than the average processing unit. In addition, the throughput is lowered at the output if out-of-band guard subcarriers and DC zero subcarriers are omitted. The sample throughput is also reduced through cyclic-prefix cancellation. This cancellation automatically happens by FFT-window selection. The reordering utilizes a dual-port memory. Therefore, it is possible to run the FFT at a higher frequency than consecutive blocks and change the clock domain inside the reordering block. If the synchronization blocks can cope with the higher clock frequency, we have two clock domains inside the FFT.

A shift operation is required to swap the right and left frequency content from natural-order to offset-order, so that the subcarrier with the most negative frequency in the Nyquist band appears first at the output. In a first version of the design, an explicit shift module was incorporated in the data processing branch to establish shifted order (cite[Max09b]). This module can be made obsolete if the input sequence $x(n)$ is multiplied with a phasor $\exp(j\pi n) = (-1)^n$, which accomplishes a frequency shift of $f_T/2$. This means that the second and forth input port of the FFT in Figure 78 must be negated. Instead of performing the DFT in Equation (302), the receiver calculates

$$X(k) = \sum_{n=0}^{N-1} (-1)^n x(n) e^{-j2\pi kn/N_{\mathrm{FFT}}} \qquad 0 \leq k \leq N-1 \tag{189}$$

IFFT operation for the transmitter merely requires that input and output samples are replaced with their complex-conjugates.

$$x(k) = \frac{(-1)^k}{N} \sum_{n=0}^{N-1} X(n) W_N^{-kn} = \overline{\frac{(-1)^k}{N} \sum_{n=0}^{N-1} \overline{X(n)}\,\overline{W_N^{-kn}}} = \frac{(-1)^k}{N} \cdot \overline{\left[\sum_{n=0}^{N-1} \overline{X(n)} W_N^{kn} \right]} \tag{190}$$

6.9 Combined de-interleaving and depuncturing

In an BICM-OFDM receiver, a de-interleaver is required between demapper and decoder. Block interleavers and de-interleavers perform a defined permutation of the input soft bit sequence. This permutation is applied for every consecutive section (or block) of the sequence. For experimental purposes, the aim had been to implement a "general-purpose" interleaver / de-interleaver, which allows a quick change of the permutation mapping. This is most easily accomplished with a memory-based design using a lookup table for the permutation rule. The de-interleaver is depicted in Figure 79. It is dedicated to one code stream.

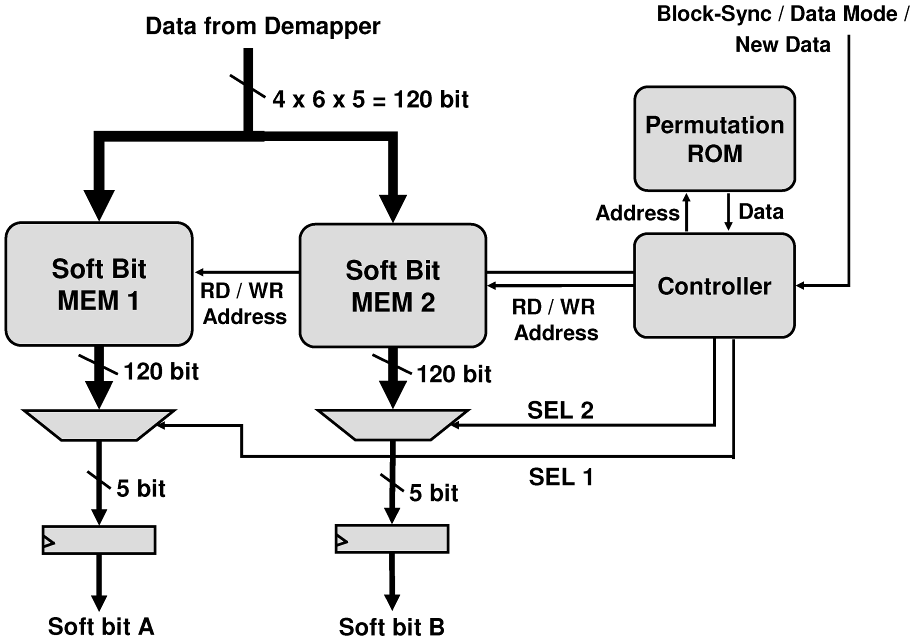


Figure 79: Simplified schematic of de-interleaver

Data is arriving from the demapper at the full bit rate on a 4-port bus. Each port carries soft bits from one demapped subcarrier per cycle. For 64-QAM, 24 soft bits arrive in parallel in each cycle. Hence, a memory width of 120 bit is required to support 64-QAM with five digits per soft bit. Data is simultaneously stored in two dual-port memory blocks to allow arbitrary access of the two output soft bits. The de-interleaver is organized like a two-page FIFO meaning that a new block can be stored while the older is being processed. A multiplexer at the output of the memory selects one out of 24 soft bits. De-interleaving mappings are stored in permutation ROM for every data mode of the PHY. This ROM is accessed by the controller and contains memory read address and multiplexer selection index for every soft bit. The de-interleaver also performs *depuncturing* for data modes using punctured convolutional encoding. This is realized with and an "out-of-range" selection index of $I = 24$. In this case, the multiplexer will output a zero for the soft bit. Very complex puncturing patterns can be realized this way. The de-interleaver supports a change of the data mode for every new block. The output rate is equal to the input rate of the Viterbi decoder, which is one message bit per cycle. Overflow may occur in the FIFO if the number of parallel de-interleavers is lower than required by the data mode.

6.10 Viterbi decoder

The basic Viterbi algorithm is explained in detail in Appendix D. In addition, it is shown in Appendix D.3 how a standard decoder can deal with bit metrics. Equations (283), (285), (286), (287), (296) and (297) are reproduced here in combined form for convenience. $G(l,s)$ denote the accumulated path metrics for state s for message bit l, $D(l,s)$ the path decisions, $s_A(s)$ and $s_B(s)$ the preceding states of state s and finally $s_{\text{pre}}(l,s)$ the chosen state $s_A(s)$ or $s_B(s)$, whichever had higher path metric for iteration l.

$$s_A(s) = \lfloor s/2 \rfloor \qquad s_B(s) = \lfloor s/2 \rfloor + M_s/2 \tag{191}$$

$$\delta(s_1, s_2, l) \in \begin{pmatrix} -m(l,1) - m(l,2) \\ -m(l,1) + m(l,2) \\ +m(l,1) - m(l,2) \\ +m(l,1) + m(l,2) \end{pmatrix} \tag{192}$$

$$G(l,s) = \max\left[G(l-1, s_A(s)) + \delta(s_A(s), s, l), G(l-1, s_B(s)) + \delta(s_B(s), s, l)\right] \tag{193}$$

$$D(l,s) = \begin{cases} 0 & \text{for} \quad G(l-1, s_A(s)) + \delta(s_A(s), s, l) > G(l-1, s_B(s)) + \delta(s_B(s), s, l) \\ 1 & \text{else} \end{cases} \tag{194}$$

$$s_{\text{pre}}(l,s) = \lfloor s/2 \rfloor + D(l,s) \cdot M_s/2 \tag{195}$$

Although there exist a lot of hardware designs as purchasable IP-cores for Viterbi decoding, it was decided to develop a (133,171) convolutional decoder with constraint length $K = 7$ dedicated to the requirements of the WIGWAM project which is also suitable for ASIC implementation. Prior to the development process is was tried to instantiate a purely register- based Viterbi decoder design on FPGA. This design does not make use of any static RAM blocks and is optimized for lower-power operation on ASIC, a detailed description is found in ([K. 07]). It became quickly clear that the utilization of dedicated RAM blocks is necessary to keep area consumption within reasonable limits.

Basically, decoding operation can be split into a trace-forward processing unit (TFP) and a trace-back processing unit (TBP). In order to achieve maximum throughput, these processors have to work in parallel. The basic memory requirements are $M_s = 64$ metric storage registers for the path metrics and the path memory. Recall that the envisaged peak data rate is much larger than clock speed and that the whole baseband needs to demultiplex the message stream into several low-rate substreams for parallel decoding. This was necessary due to the fact that the highest parallelization achieved with a traditional decoder design is to implement M_s add-compare-select units in parallel. Since the ACS operation is nonlinear and recursive in nature, pipelining cannot be used. In [G.F89], the attempt has been made to break the bottleneck of the ACS-operation by introducing an m-step trellis operation. While sounding promising, the processing gets quickly involved with increased step size and the promised big performance enhancement might not be achieved in an real implementation. Considering time limitations for design development, it has been decided to follow the approved approach of conventional ACS implementation.

We assume that the ACS units will perform their operations within a single clock cycle, so that the decoder is able to process one message bit per cycle. The trace-forward processor consists of a preprocessor and the ACS machine. The preprocessor calculates the four metric combinations given in Equation (192). These are used as increments for the path metrics in the ACS machine. To account for the high fanout, another register stage has been inserted between preprocessor and ACS machine, doubling the required number of TFP registers. On the other hand, these intermediate registers have shorter bitwidth compared to the ACS accumulators (6 bit compared to 10 bit).

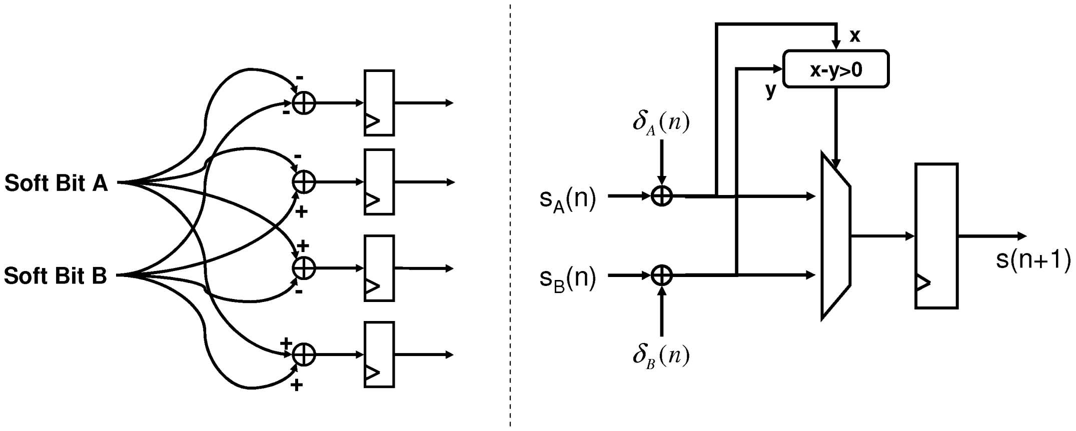


Figure 80: Preprocessing stage and one out of 64 ACS units

The ACS operation for state s consists in first adding path metric increments to the metrics of precedent states, which lead to the new state, then comparing which of the two incremented metrics is higher and finally selecting that metric as the updated path metric of state s (Equations (193) and (194). This happens for M_s = 64 parallel units and the output of the ACS machine is a 64 bit wide decision vector per iteration. This bitvector need to be stored in a 64-bit wide path memory. The instantiated design for the WIGWAM demonstrator uses 5 bits to represent soft bit metrics. With wrap-around arithmetic used for the ACS units, 10 bits are found to be sufficient for the accumulated path metrics.

Now the traceback operation is considered. It must be noted that the decoder has been implemented as a fully synchronous block featuring one clock domain. During each iteration n, the following steps have to be performed:

1. Read out the stored path metrics for the current message bit l
2. Select the path decision $D(l, s_n)$ out of all path M_s decisions for the current state s_n
3. Determine the next state s_{n+1} according to equation (195)
4. If not finished, perform the same operations for message bit $l-1$ during the next iteration $n+1$

As described in Appendix D, the decoder must perform at least $\Delta_m = 5 \cdot K$ backsteps before output bits can be reliably obtained. Hence, the traceback procedure has been divided into two phases. During the first phase, N_{idle} backsteps are performed creating no output, whereas one output bit is obtained for each backstep during phase 2. We denote the number of retrieved message bits during a normal traceback procedure as N_{msg}. The implementation performs each backstep in a single cycle. Since N_{msg} bits are read out within a period of $N_{\text{idle}} + N_{\text{msg}}$ cycles, the average throughput D_{tr} of a single traceback unit is lower than one bit per cycle and therefore lower than required.

$$D_{\text{tr}} = \frac{N_{\text{msg}}}{N_{\text{msg}} + N_{\text{idle}}} \tag{196}$$

Therefore, the implementation uses two traceback units (TRU) in parallel. By choosing $N_{\text{idle}} = N_{\text{msg}}$, each TRU has an average throughput of $D_{\text{tr}} = 1/2$ bits per cycle, so that both TRUs together just meet the requirements.

Recall that the coded message stream is terminated with zero tail bits, and that the last traceback procedure has to read out all residual message bits stored in path memory. Hence, the last traceback has no idle phase. The regular flow chart is depicted in Figure 81, assuming a continuous input code stream without interruptions.

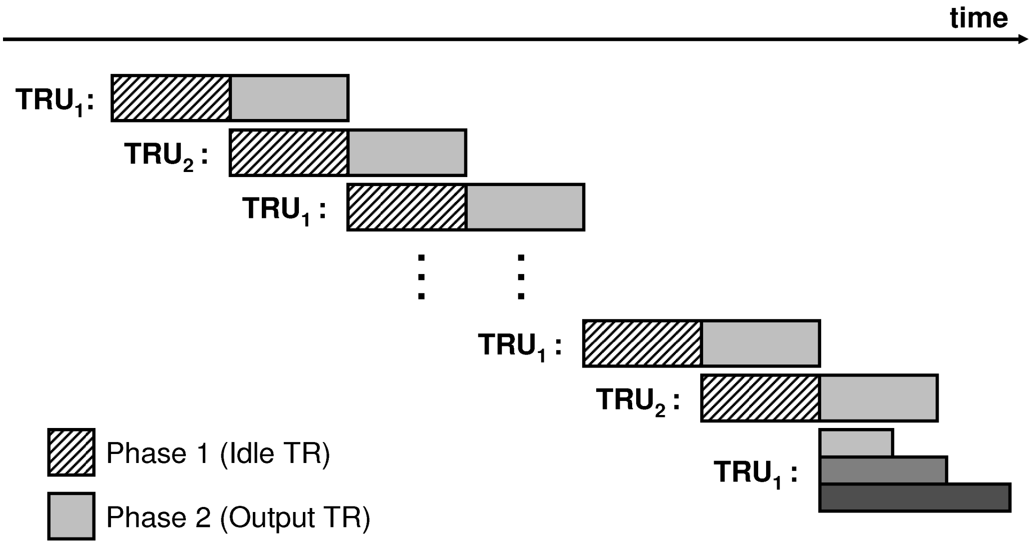


Figure 81: Regular flow chart of traceback processing units for continuous input

Special care had to be taken to ensure correct data flow. Note that the last traceback delivers data bits simultaneously with the previous traceback and, depending on the packet length, may finish before or after the previous one.

The block diagram of the full design is shown in Figure 82. The decoder features data-driven processing of the input in order to tolerate non continuous input streams. The whole architecture ensures correct data flow and continuous operation without the need for input wait cycles. In addition, the input data stream can be marked as terminated using a termination signal and nonetheless continue with the next cycle.

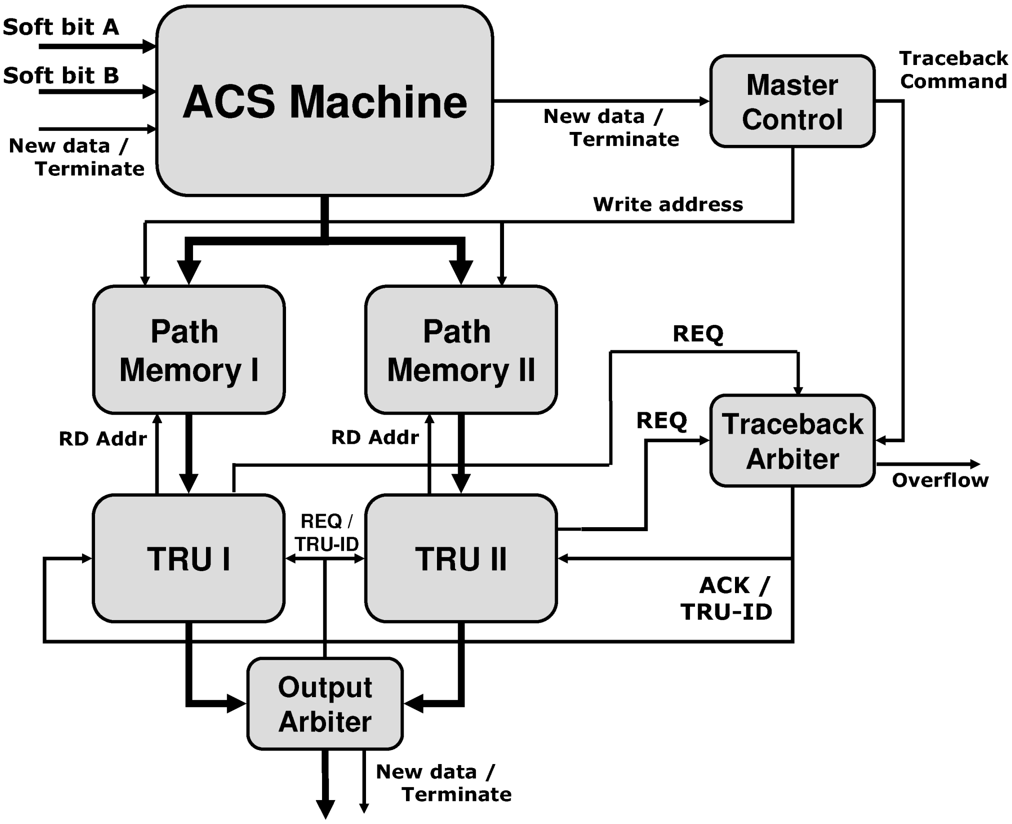


Figure 82: Viterbi decoder block diagram

The ACS machine will produce one slice of path decisions and update state metrics for every valid dual soft bit input. It also will signalize a new data bit to the *master controller (MC)* together with the termination information. The path decisions are written into two path memory blocks, which carry the same information. These blocks are dual-port RAMs, and each of them is read out by one of the traceback units. The master controller increments the write address for the path information, but its main task is to create traceback commands (TRC) for the TRUs. Each command consists of the read address for the first traceback position, the number of traceback steps, the number of valid output bits and the termination bit.

These commands enter a command-FIFO with a depth of four in the *traceback arbiter (TRA)*. The traceback commands are fetched by the TRUs after completion of the previous traceback cycle. In order to avoid data confusion, the TRA will never assign two consecutive TRCs to the same TRU. Such a case can happen when the stream is terminated and new input data directly follows. For the same purpose, an *output arbiter* is inserted, which alternately fetches data bursts from the TRUs.

To be able to create traceback commands, the master controller keeps track of the number of stored message bits in the path memory. Its behavior is shown in Figure 83. The complete flow chart is executed for every cycle. N_{message} is the number of stored message bits. During a regular traceback procedure, N_{TR} backsteps are performed to obtain N_{OUTPUT} message bits. In relation to the previous notation we have $N_{\text{OUTPUT}} \equiv N_{\text{msg}}$ and $N_{\text{TR}} \equiv N_{\text{msg}} + N_{\text{idle}}$. In the implementation, $N_{\text{OUTPUT}} = 48$ and $N_{\text{TR}} = 96$ yield the required throughput of one bit per cycle and ensure a minimum traceback depth of about $7K$. If the stream is terminated, a TRC will immediately be initiated for all remaining stored message bits.

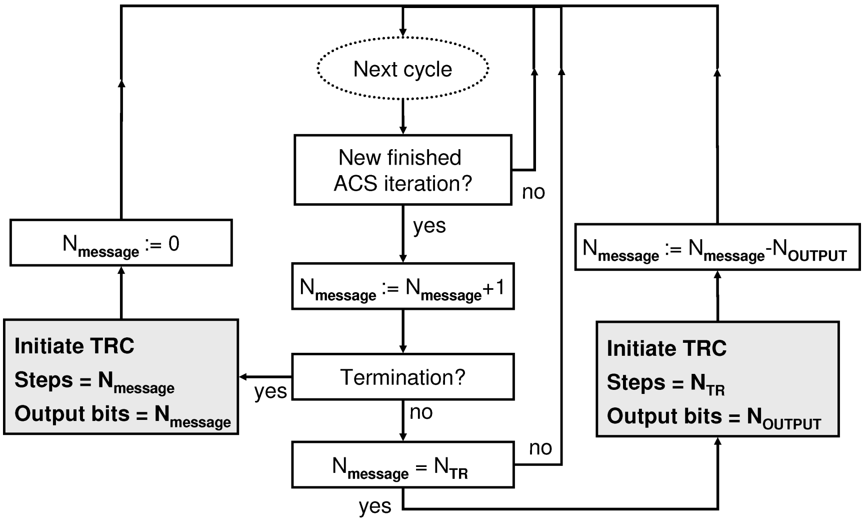


Figure 83: Viterbi master controller

The internal structure of the TRU is shown in Figure 84. The *path calculation unit* mainly consists of a 64:1 multiplexer to select the current path decision from $M_s = 64$ decisions arriving from path memory each clock cycle, Equation (194). The current state variable s_n is taken as the index variable. The next state is obtained by shift operation and decision about the MSB (Equation (194)). The message data is given by the LSB of the current state. Because data is obtained in reverse order, two shift registers are used to deliver data in correct order to the output.

During normal continuous operation, the TRU behaves as follows. Each retrieved message bit is first shifted in a $N_{\text{TR}} = 96$ long shift register SR1. After the last bit is pushed, the register is entirely copied into a second shift register SR2, which is then shifted out with 16 bits per cycle. At the end of the current traceback, the TRU sends a request for the next TRC just in the right cycle to avoid idle cycles. Otherwise, the throughput condition would be violated.

The TRU can only send a new request if SR2 is empty and ready to take the next data burst. Idle cycles only occur in two cases. A new request might be not immediately acknowledged by the TRA sending a new TRC command. Also, data might get stuck in SR2, if the output arbiter has not yet fetched it. Both cases can only occur at stream termination where internal processing differs from the regular case. Although it might seem that these idle cycles lead to data loss, this is not the case. At first, note from Figure 81 that traceback throughput is temporarily doubled at stream termination. Hence, more data is buffered in the TRUs than usual and idle cycles can be tolerated. The path memory is made up of FPGA block rams with a minimum depth of 512 words. It is organized like FIFO memory with circular addressing. Before unprocessed data path could be overwritten, the TRC FIFO would already experience an overflow. Simulation shows that even with extremely short packets of only one byte and another zero termination byte, no overflow occurs.

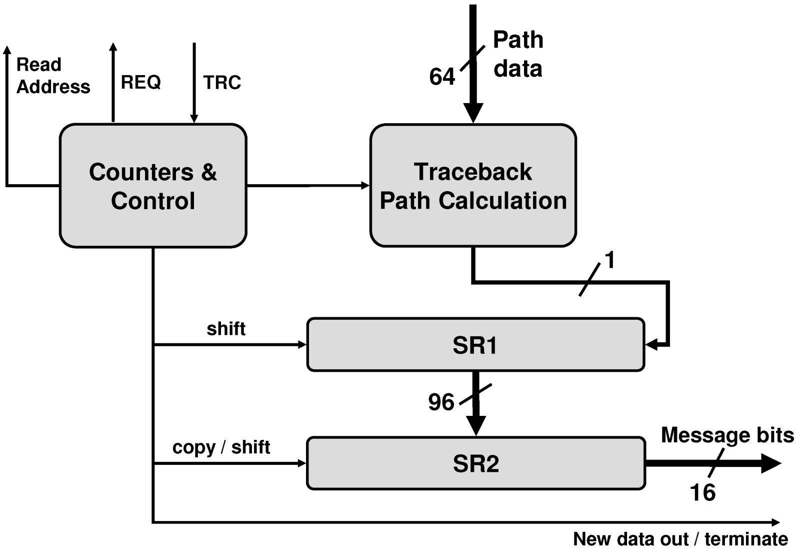


Figure 84: Traceback Unit

The synthesis tool reports a frequency of 120 MHz after routing for the Virtex-2 Pro VP 100 device. The number of occupied slices is around 4%. Interestingly, the critical path, which sets the speed limit, is not the ACS calculation, but made up by the one-cycle based path calculation in the TRUs. In principle, a higher speed could be obtained by doubling the number of TRUs. Four TRUs can perform two-cycle based traceback operation and achieve the same throughput. This requires two additional path memory blocks and also leads to higher decoder latency.

6.11 Digital automatic gain control

As mentioned in the overview section, a digital AGC has been incorporated for signal leveling. This block usually consists of a power detector, a low-pass (loop) filter together with a control signal generator and a variable gain amplifier (VGA).

Basically, there are two AGC topologies common, the feed-forward AGC and feedback AGC as shown in Figure 85. For the present case, the feed-forward type seems to have several advantages. The expected high pipeline latency in the detector-filter-control path imposes restrictions on the filter bandwidth and therefore may limit the speed of the filter. In contrast, the feed-forward scheme is always stable if only the loop filter is unconditionally stable. Furthermore, high latency in the control path can be compensated with an introduced delay-element in the signal path preceding the VGA. We can make

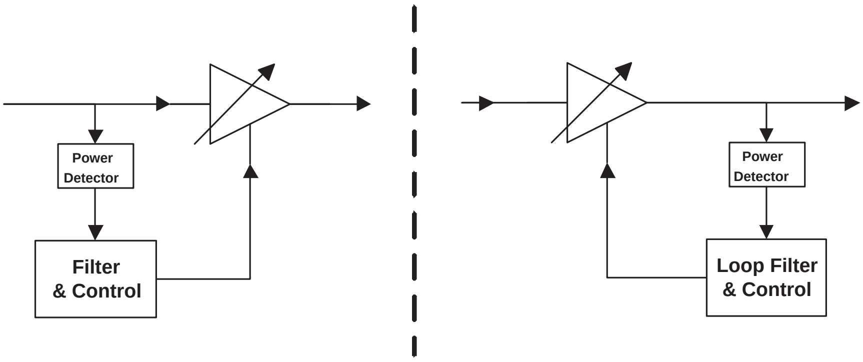


Figure 85: Feed-Forward and Feedback AGC topology

the following *ansatz* for the input-output signal relationship

$$z_{\mathrm{out}}(n) = z_{\mathrm{in}}(n) \cdot \sqrt{P_{\mathrm{ideal}}/\widetilde{P}_{\mathrm{avg}}(n)} \tag{197}$$

In Equation (197), $z_{\mathrm{in}}(n)$ and $z_{\mathrm{out}}(n)$ represent the input and output complex baseband signal, $\widetilde{P}_{\mathrm{avg}}(n)$ denotes the continuously estimated average power at time index n, and P_{ideal} the desired output power level. With this approach, the main task consists in estimation of some form of "averaged power" $\widetilde{P}_{\mathrm{avg}}(n)$.

We assume a hardware with a clock frequency f_{clock} and sampling frequency $f_T = N \cdot f_{\mathrm{clock}}$ so that N signal ports need to be processed in parallel. A first averaged power value $P_D(n)$ (with suffix D for "detector") can be obtained as

$$P_z(n) := |z(n)|^2 \tag{198}$$

$$P_D(n) := \frac{1}{K} \sum_{k=0}^{K-1} P_z(n-k) \tag{199}$$

It is well known that the moving average operation in (199) can be efficiently implemented using the recursive formula[1]

$$P_D(n) = P_D(n-1) + \frac{1}{K} P_z(n) - \frac{1}{K} P_z(n-K) \tag{200}$$

Equation (200) is a bit cumbersome to evaluate for an parallel N-port system, because it is required to unroll this equation for all ports. This may lead to timing issues. N-ary successive application of (200) yields

$$P_D(n) = P_D(n-N) + \frac{1}{K} \sum_{k=0}^{N-1} P_z(n-k) - \frac{1}{K} \sum_{k=0}^{N-1} P_z(n-K-k) \tag{201}$$

A simplification is achieved if the averaged power $P_D(n)$ is *downsampled* by a factor of N.

$$\widehat{P}_D(m) := P_D(n = Nm) \tag{202}$$

---

[1] K is chosen as a power of 2 to have shifting instead of division operation

Justification for this downsampling will be provided later. From (201) and (202) it follows that

$$\widehat{P}_D(m) = \widehat{P}_D(m-1) + \frac{1}{K}\left[\sum_{k=0}^{N-1} P_z(mN-k)\right] - \frac{1}{K}\left[\sum_{k=0}^{N-1} P_z(mN-K-k)\right] \tag{203}$$

If K is chosen as an integer multiple of N, $K = LN$, we obtain from eq. (203)

$$\begin{aligned}\widehat{P}_D(m) &= \widehat{P}_D(m-1) + \frac{1}{K}\left[\sum_{k=0}^{N-1} P_z(mN-k)\right] - \frac{1}{K}\left[\sum_{k=0}^{N-1} P_z((m-L)N-k)\right] \quad (204)\\ &= \widehat{P}_D(m-1) + Q(m) - Q(m-L)\end{aligned}$$

$$Q(m) := \frac{1}{K}\sum_{k=0}^{N-1} P_z(mN-k) \tag{205}$$

From (204) and (205) it follows that downsampling *after* moving-average operation is equivalent to the following operations: addition of N consecutive power values $P_z(n)$ followed by downsampling by a factor of N and conventional single-port moving-average operation for L elements.

In an FPGA design, a single memory block comes at low cost and can be dedicated to the AGC unit (see appendix I). In order to avoid amplitude modulation of the AGC on the data frame, the AGC gain value can be fixed by the synchronizer as soon as the frame is detected. Hence, in can be maintained that the AGC will only operate on the first preamble part. The waveform of this part consists of a periodic sequence with period N_D. Power averaging can be performed over one or several periods choosing $K = lN_D$ for some integer l. In this case, the true average power would determined for the *noise free* signal waveform after the initial guard time and K samples have passed. The output of the sliding averager is used as the estimate for the mean power in (197).

$$\widetilde{P}_{\text{avg}}(n) := \widehat{P}_D(m = \lfloor n/N \rfloor) \tag{206}$$

The only effect of the downsampling is that the estimated average power and hence the gain value is updated only for every N-th sample. This does hardly influence system performance. We rewrite Equation (197) as (207) to obtain a convenient implementation for square-root operation and division. For this purpose, processing changes to base-2 logarithm domain and back.

$$z_{\text{out}}(n) = z_{\text{in}}(n) \cdot 2^{(\log_2[P_{\text{ideal}}] - \log_2[\widetilde{P}_{\text{avg}}(n)])/2} \tag{207}$$

In an ASIC implementation, a dedicated memory block or long shift register might not be desirable for its higher area consumption. An alternative solution is to introduce a first-order filter in the signal chain of the form

$$y(m) = (1-\alpha)\cdot y(m-1) + \alpha \cdot x(m-1) \tag{208}$$

α stands for the *forgetting factor* and can be conveniently chosen as $\alpha = 2^{-v}$. The question arises where to place this element in the chain. This can be done in the linear domain with $x = \widehat{P}_D(m)$ or log-domain with $x = log_2(\widehat{P}_D(m))$. A generic model with all elements is shown in Figure 86. Each stage can be bypassed by setting either $L = 1$, $\alpha_1 = 1$ or $\alpha_2 = 1$. Note that the reduced sampling rate

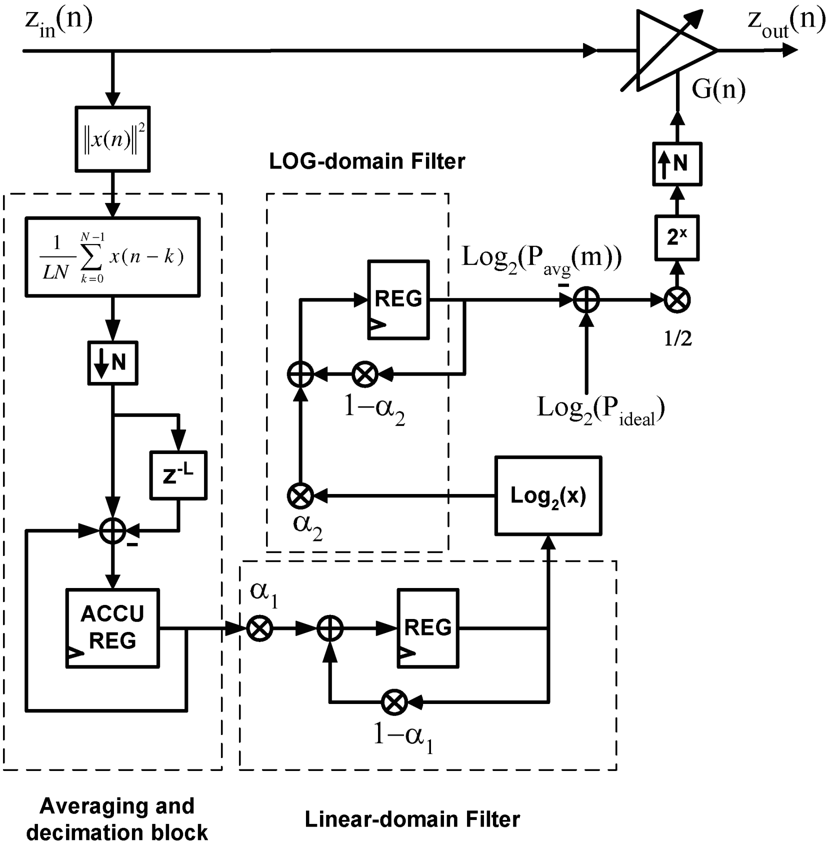


Figure 86: Generic AGC block diagram

f_T/N slows down the filter-elements by a factor of N, but the initial averaging in (205) also provides some reduction of the power fluctuations.

We consider the linear-domain filter and examine its system response for a step change at the input from some fixed power level $\widehat{P}_D(m) = P_s$ for $m < 0$ to some other fixed power level $\widehat{P}_D(m) = P_e$ for $m \geq 0$. The result is (with $P_F(n) \equiv y(n)$)

$$P_F(m) = \begin{cases} (1-\alpha_1)^m(P_s - P_e) + P_e & \forall m \geq 0 \\ P_s & \forall m < 0 \end{cases} \tag{209}$$

Now the required sample time m_e shall be calculated to obtain an error less than $\Delta P = \beta_1 P_e$.

$$\begin{aligned} \Delta P = |P_F(m_e) - P_e| &\leq \beta_1 P_e \Leftrightarrow (1-\alpha_1)^{m_e} \cdot |P_s - P_e| \leq \beta_1 P_e \\ \Leftrightarrow m_e &\geq \frac{\log_{10}(|P_s - P_e|/(\beta_1 P_e))}{\log_{10}(1/(1-\alpha_1))} \Leftrightarrow m_e = \lceil \gamma_1 \log_{10}(|P_s - P_e|/(\beta_1 P_e)) \rceil \end{aligned} \tag{210}$$

with $\gamma_1 = 1/\log_{10}(1/(1-\alpha_1)) > 0$. For the frame start we may assume $P_e >> P_s$ and for the frame end $P_e << P_s$. This yields

$$\begin{aligned} m_e &\approx \lceil \gamma_1 \log_{10}(1/\beta_1) \rceil & \text{for} \quad P_e >> P_s \\ m_e &\approx \lceil \gamma_1 \log_{10}((P_s/P_e) \cdot 1/\beta_1) \rceil = \left\lceil (\tfrac{\gamma_1}{10}) \cdot \mathrm{SNR}_{\mathrm{dB}} + \gamma_1 \log_{10}(1/\beta_1) \right\rceil & \text{for} \quad P_e << P_s \end{aligned} \tag{211}$$

with $\mathrm{SNR}_{\mathrm{dB}} = 10\log_{10}(P_s/P_e)$. Therefore, the settling time is faster for the ramp-up case ($P_e >> P_s$) at the beginning of the frame compared to the the ramp-down case at the frame end. A symmetrical behavior is achieved if the filter works in log-domain ($x = log_2[\widehat{P}_D(m)]$). Equations (209) and (210) are replaced with (212) and (213) (with $\gamma_2 = 1/\log_{10}(1/(1-\alpha_2))$).

$$y(m) = \begin{cases} (1-\alpha_2)^m(log_2(P_s) - log_2(P_e)) + log_2(P_e) & \forall m \geq 0 \\ log_2(P_s) & \forall m < 0 \end{cases} \tag{212}$$

$$\begin{aligned} |y(m) - log_2(P_e)| &\leq \beta_2 \\ \Leftrightarrow m_e &= \lceil \gamma_2 \log_{10}(|log_2(P_s) - log_2(P_e)|) + \gamma_2 \log_{10}(1/\beta_2) \rceil \\ &= \lceil \gamma_2 \log_{10}(\mathrm{SNR}_{\mathrm{dB}}) + \gamma_2 \log_{10}[1/(10\log_{10}(2)\beta_2)] \rceil \end{aligned} \tag{213}$$

The settling time with respect the original sampling rate is given to

$$n_e = N \cdot m_e \tag{214}$$

- For an error of less than 0.5 dB, $\beta_1 = 0.108$ and $\beta_2 = 0.166$.
- In practice, the exponents $log_2[\widehat{P}_D(m)]$ must be limited to a most negative value $E_{\min}$ for very small $\widehat{P}_D(m)$, i.e. $log_2[\widehat{P}_D(m)]$ is replaced by $\max(E_{\min}, log_2[\widehat{P}_D(m)])$.
- The log-domain filter actually cannot work without the preceding averaging stage. Without, small instantaneous power values $\widehat{P}_D(m)$ would cause large fluctuations of $log_2[\widehat{P}_D(m)]$.

When the AGC is applied to the preamble, a steady state is reached after the initial transient from noise level to signal level. This steady state lasts for the duration of the first preamble part. The amount of residual amplitude modulation depends on the noise level, the choice of filter and the forgetting factor α. At the moment when the synchronizer detects the frame, the current AGC gain value is frozen, leading to some constant residual gain error for the current frame.

Table 12: AGC parameters

Type	N	L	α_1	α_2	E_{min}
Averager (NB)	4	32	1	1	-10
Linear-domain filter (NB)	4	1	2^{-4}	1	-10
Log-domain filter (NB)	4	4	1	2^{-4}	-10
Averager (WB)	8	32	1	1	-10
Linear-domain filter (WB)	8	4	2^{-3}	1	-10
Log-domain filter (WB)	8	4	1	2^{-3}	-10

The three different AGC topologies have been compared by simulation and shown in Figure 87 for both OFDM modes. The AGC parameters (see table) have been optimized to achieve the fastest response for a given maximum "outage probability" of the gain error. Since the OFDM system will hardly work below 3 dB SNR, the goal has been to attain a probability of less than 0.1% for the gain error to exceed 1 dB at an SNR of 3 dB. The settling behavior is shown in Figure 88 for the 400 MHz mode in the upper graph and the 2160 MHz mode in the lower graph. The SNR was set to 40 dB to provoke the longest settling time[1]. The averager delivers best performance for both modes. Settling requires two A-sequences (or guard times) (= 320 ns) for the narrowband and one A-sequence (= 118 ns) for the wideband mode.

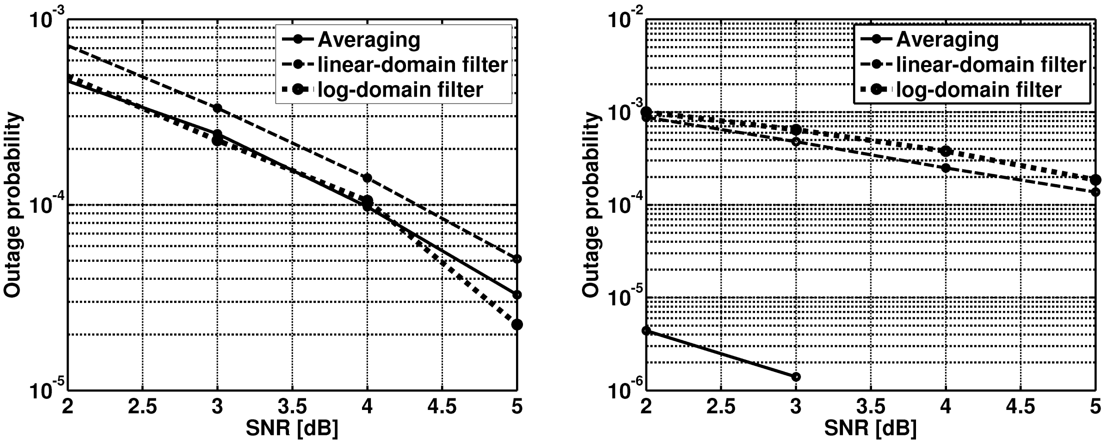


Figure 87: Outage probability for 1 dB gain error for narrowband (left) and wideband mode (right))

The log-domain filter has the longest settling time for the narrowband mode and should not be used. For the wideband mode, it shows better behavior in comparison, since it reaches a 1 dB error roughly at the same time with the linear-domain filter.

Table 13: Calculated rise and fall times for 0.5 dB error

Type	Rise time in samples (ns)	Fall time in samples (ns)
Averager (NB)	$\approx$ 128 (320 ns)	$\approx$ 128 (320 ns)
Linear-domain filter (NB)	138 (345 ns)	709 (1.77 μs)
Log-domain filter (NB)	271 (677.5 ns)	271 (677.5 ns)
Averager (WB)	$\approx$ 256 (118.5 ns)	$\approx$ 256 (118.5 ns)
Linear-domain filter (WB)	133 (61.5 ns)	685 (317 ns)
Log-domain filter (WB)	263 (122 ns)	263 (122 ns)

As predicted, the linear-domain filter has a long decay time at the frame end. This decay time measures roughly 2 OFDM symbols (2 x 320 samples) or 1.6 microseconds for the narrowband and one OFDM symbol (around 1300 samples) or 600 ns for the wideband mode. It is easy to speed-up this behavior by temporarily changing the update factor α_1. With this in mind, we can conclude that the linear-domain filter has better noise performance compared to the log-domain filter and therefore achieves a faster settling. Note that simulation was performed without channel delay spread. To account for channel effects, the settling time extends up to one guard time.

---

[1] A higher SNR is unlikely to appear.

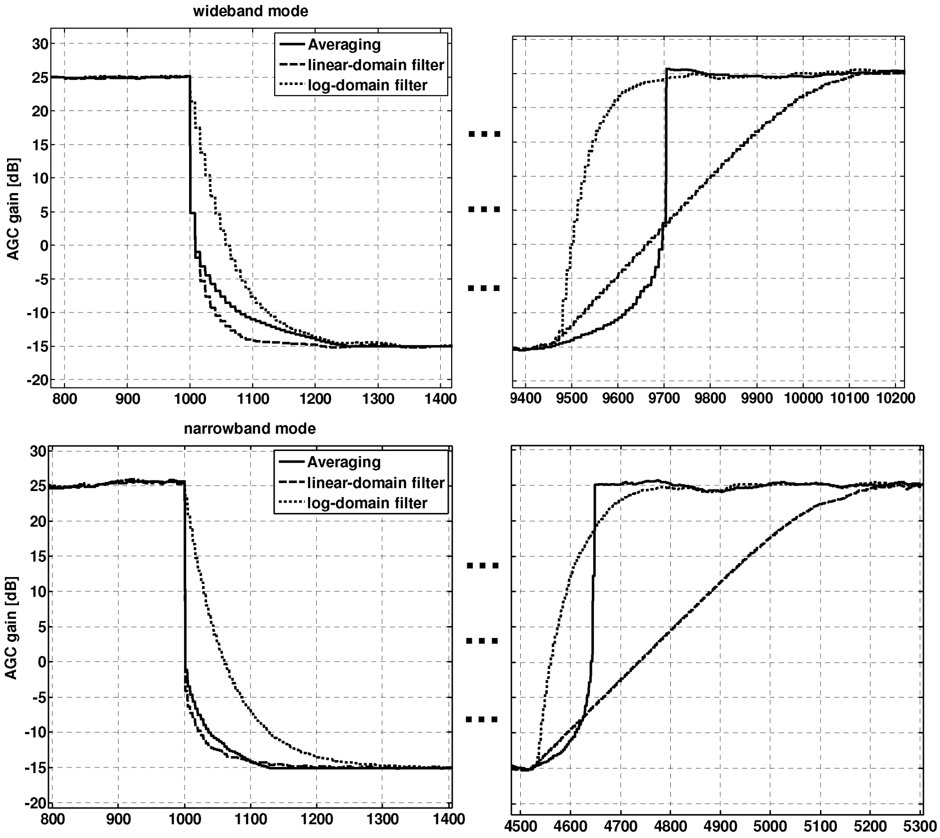


Figure 88: AGC settling for SNR = 40 dB

The calculated settling times using Equations (211),(213) and (214) are given in table 13. Although the calculation is done for a step response, the resulting values turn out to be good approximations to the simulation results.

7 Conclusion

In this thesis, the suitability of OFDM transmission has been analyzed under ’real’ conditions. These conditions have a temporary character as they refer to RF components, which are currently available. We have seen that the typical shortcomings of OFDM are especially encountered at 60 GHz. Phase noise of integrated oscillators always leads to some performance degradation in the order of 0.5-1 dB. The used power amplifier model with a graceful intersection between the linear and the limiter region requires a high backoff. Therefore, antenna gain is required to overcome link-budget limitations. It was highlighted that the preamble length is rather determined by phase noise leading to degraded CFO estimation. For this purpose, an exact analytical expression for the phase-noise dependent CFO error was calculated and compared with simulation. Furthermore, an expression was found to quantify the created interference by insufficient guard time extension, although it seems unlikely that this equation has not been found previously. Based on the goal to balance the influence of these degradation sources, OFDM PHY parameters have been chosen.

The baseband specification was driven by the hardware limitations of the FPGA platform calling for a high degree of parallelization and pipelining. These restrictions forbid the application of feedback techniques, since they would suffer from high latency. Nevertheless, it was possible to design an almost pure feed-forward system showing good performance. The developed new synchronization algorithm achieves robust detection performance at virtually zero false alarm rate. The extended preamble length calls for the frame aggregation technique, which in turn requires to keep track of the channel. It was shown that the system provides reasonable performance even with a suboptimum channel re-estimation method based on the sample-and-hold principle. On the other hand, this technique comes at virtually no additional hardware effort, since it can be covered by the channel estimator. The tracking of phase and timing has undergone an evolution from the WIGWAM to the EASY-A system and finally achieves robust performance. It was shown how to organize the receiver algorithms to ensure a flawless coexistence of channel tracking and carrier/timing-tracking, despite the latency issues.

To reduce the hardware effort, cross-correlation based timing synchronization has been avoided in favor of a phase unwrapping method showing sufficient performance for the applied radio channel models. A very efficient hardware implementation has been found to reuse blocks for the aim of channel estimation and time synchronization.

The wideband PHY offers a number of advantages compared to the first system. A simple noise reduction technique was applied to achieve a high estimation gain, which enables a reduction of the preamble length. An improved interleaving scheme was found offering an additional design parameter to outperform the 802.11a standard interleaver. Application of the RS outer encoder has turned out to be very essential for the highest data modes, which require puncturing. On the other hand, it was demonstrated that puncturing fails in non-line-of-sight channel conditions. In addition, soft bits for the Viterbi decoder require more than 5 bit, if degradation shall be kept at minimum. The RS-convolutional scheme has been compared with an established LDPC code showing slightly better performance. In such a case, OFDM can offer its advantage. During the project, analog components significantly improved in terms of phase noise and output power whereas the radio channel keeps being the same.

Hence, we can assume a general technology development in favor of the OFDM technique.

A Derivation of the phase-noise induced CFO estimation performance

In section 4.4, the phase noise induced estimation error is considered. The error variance is derived in the following using the Wiener oscillator model. We reprint Equations (84) and (85) for convenience.

$$\begin{aligned} z_{CH}(t) &= z(t) \otimes h(\tau) \qquad (215) \\ z_{RX}(t) &= z_{CH}(t) \cdot \exp(j(2\pi \Delta f t + \phi(t))) + \eta(t) \end{aligned}$$

$$v = \frac{1}{T_I} \int_{t=0}^{T_I} \overline{z_{RX}(t)} z_{RX}(t + T_\Delta) dt \qquad (216)$$

$$T_a = T_I + T_\Delta \qquad (217)$$

Equation (215) is the signal model where we set the additive noise component to $\eta(t) = 0$. (216) defines the considered estimator for integration interval T_I and step size T_Δ, and (217) just tells that the integration interval T_I reduces with rising step interval T_Δ for fixed observation interval T_a. We assume a larger integration interval than the step size, $T_I \geq T_\Delta$. The input signal z_{CH} is periodic over T_Δ, $z_{CH}(t) = z_{CH}(t + T_\Delta)$ for $0 \leq t \leq T_I$. Combining (215) and (216), using and the small signal approximation $exp(x) \approx 1 + x$ gives

$$\begin{aligned} v &= \frac{1}{T_I} \int_{t=0}^{T_I} \overline{z_{RX}(t)} z_{RX}(t + T_\Delta) dt \qquad (218) \\ &= \exp(j2\pi \Delta f T_\Delta) \frac{1}{T_I} \int_{t=0}^{T_I} |z_{CH}(t)|^2 \exp(j[\phi(t + T_\Delta) - \phi(t)]) dt \\ &\approx \exp(j2\pi \Delta f T_\Delta) \left[\frac{1}{T_I} \int_{t=0}^{T_I} |z_{CH}(t)|^2 dt + j \frac{1}{T_I} \int_{t=0}^{T_I} |z_{CH}(t)|^2 [\phi(t + T_\Delta) - \phi(t)] dt \right] \\ &= \exp(j2\pi \Delta f T_\Delta) \left[P_{z_{RX}} + j \frac{1}{T_I} \int_{t=0}^{T_I} |z_{CH}(t)|^2 [\phi(t + T_\Delta) - \phi(t)] dt \right] \end{aligned}$$

It can be easily shown using (3) that the phase difference $\Delta\phi(t, t + T_\Delta) = \phi(t + T_\Delta) - \phi(t)$ follows again a Wiener process with linearly increasing variance $\sigma_{\Delta\phi}(T_\Delta)^2 = 2DT_\Delta$, which depends only on the relative time difference T_Δ. Obviously, $\sigma_{\Delta\phi}(T_\Delta) = \sigma_\phi(t = T_\Delta)$, and the approximation in (218) is good if the step width is chosen small enough to ensure $k\sigma_\phi(T_\Delta) = k\sqrt{2DT_\Delta} << 1$ for approximately $k \geq 5$. Since $\arctan(x) \approx x$ for $|x| << 1$, the angle of v is approximately equal to

$$(\angle v) = \arctan(\mathrm{Im}\{v\}/\mathrm{Re}\{v\}) \approx 2\pi \Delta f T_\Delta + \frac{1}{T_I} \int_{t=0}^{T_I} |z_{CH}(t)|^2 [\phi(t + T_\Delta) - \phi(t)] dt \qquad (219)$$

Setting $g(t) := |z_{CH}(t)|^2$, the frequency estimation error reads

$$\Delta f_{\text{err}} = \frac{(\angle v) - 2\pi \Delta f T_\Delta}{2\pi T_\Delta} \approx \frac{1}{2\pi T_\Delta T_I P_{z_{RX}}} \int_{t=0}^{T_I} g(t) [\phi(t + T_\Delta) - \phi(t)] dt \qquad (220)$$

Because of (4), $E\left\{\Delta f_{\text{err}}\right\} = 0$. Hence the estimator is unbiased. The variance is written as

$$\sigma^2_{\Delta f_{\text{err}}} = E\{(\Delta f_{\text{err}})^2\} = E\left\{ \frac{1}{(2\pi T_\Delta T_I P_{z_{RX}})^2} \left(\int_{t=0}^{T_I} g(t) [\phi(t + T_\Delta) - \phi(t)] dt \right)^2 \right\} \qquad (221)$$

If there are more than two sequences ($N_s > 2$), the periodicity $g(t) = g(t + T_\Delta)$ for $0 \leq t \leq T_I$ leads to cancellation of all contributions in the integral for which $T_\Delta \leq t \leq T_I$. The following two terms remain, which is also valid for $N_s = 2$:

$$\begin{aligned}
\sigma^2_{\Delta f_{\mathrm{err}}} &= \frac{1}{(2\pi T_\Delta T_I P_{z_{RX}})^2} E\left\{\left(\int_{t=T_I}^{T_I+T_\Delta} g(t)\phi(t)dt - \int_{\alpha=0}^{T_\Delta} g(\alpha)\phi(\alpha)d\alpha\right)^2\right\} \qquad (222)\\
&= \frac{1}{(2\pi T_\Delta T_I P_{z_{RX}})^2}[E\left\{\int_{\alpha,t=0}^{T_\Delta} g(t)g(\alpha)\phi(t+T_I)\phi(\alpha+T_I)d\alpha dt\right\}\\
-2E&\left\{\int_{\alpha,t=0}^{T_\Delta} g(t)g(\alpha)\phi(t+T_I)\phi(\alpha)d\alpha dt\right\} + E\left\{\int_{\alpha,t=0}^{T_\Delta} g(t)g(\alpha)\phi(t)\phi(\alpha)d\alpha dt\right\}]\\
&= \frac{1}{(2\pi T_\Delta T_I P_{z_{RX}})^2}[\int_{\alpha,t=0}^{T_\Delta} E\left\{g(t)g(\alpha)\right\} R_{\phi\phi}(t+T_I, \alpha+T_I)d\alpha dt\\
-2&\int_{\alpha,t=0}^{T_\Delta} E\left\{g(t)g(\alpha)\right\} R_{\phi\phi}(t+T_I, \alpha)d\alpha dt + \int_{\alpha,t=0}^{T_\Delta} E\left\{g(t)g(\alpha)\right\} R_{\phi\phi}(t, \alpha)d\alpha dt]\\
&= \frac{1}{(2\pi T_\Delta T_I P_{z_{RX}})^2}\left[A - 2B + C\right]
\end{aligned}$$

For the Wiener process, $R_{\phi\phi}(t,\tau) = 2D\min(t,\tau)$. One can calculate the expected variance for a specific channel setting $E\left\{g(t)g(\alpha)\right\} \equiv g(t)g(\alpha)$. If on the other hand the channel is treated as a random variable, knowledge about $E\left\{g(t)g(\alpha)\right\}$ is required, where $g(t)$ and $g(\alpha)$ are the squared magnitudes of the received at time t and α. Moreover, the receiver may evaluate the performed autocorrelation not exactly at the ideal timing but within some margin before or after. This behaviour "smeares" the statistical value $E\left\{g(t)g(\alpha)\right\}$. For a qualitative statement, it may be sufficient to simply assume

$$E\left\{g(t)g(\alpha)\right\} = (P_{z_{RX}})^2 \qquad (223)$$

This assumption is equivalent to the case of transmitting a constant-envelope preamble signal over a frequency-flat channel. Calculation of the integrals A, B and C and finally the CFO error variance $\sigma^2_{\Delta f_{\mathrm{err}}}$ is straight forward.

$$\begin{aligned}
C &= \int_{\alpha,t=0}^{T_\Delta} (P_{z_{RX}})^2 R_{\phi\phi}(t,\alpha)d\alpha dt = (P_{z_{RX}})^2 \cdot 2D \cdot \int_{\alpha,t=0}^{T_\Delta} \min(t,\alpha)d\alpha dt \qquad (224)\\
&= (P_{z_{RX}})^2 \cdot 2D \cdot \int_{t=0}^{T_\Delta}\left(\int_{\alpha=0}^{t} \alpha d\alpha + \int_{\alpha=t}^{T_\Delta} t d\alpha\right) dt = (P_{z_{RX}})^2 \cdot 2D \cdot \frac{1}{3}T_\Delta^3\\
A &= \int_{\alpha,t=0}^{T_\Delta} (P_{z_{RX}})^2 R_{\phi\phi}(T_I+t, T_I+\alpha)d\alpha dt\\
&= (P_{z_{RX}})^2 \cdot 2D \cdot \int_{\alpha,t=0}^{T_\Delta} [T_I + \min(t,\alpha)]d\alpha dt = (P_{z_{RX}})^2 \cdot 2D \cdot \left(T_I T_\Delta^2 + \frac{1}{3}T_\Delta^3\right)\\
B &= \int_{\alpha,t=0}^{T_\Delta} (P_{z_{RX}})^2 R_{\phi\phi}(T_I+t, \alpha)d\alpha dt\\
&= (P_{z_{RX}})^2 \cdot 2D \cdot \int_{\alpha,t=0}^{T_\Delta} \min(t+T_I, \alpha)d\alpha dt = (P_{z_{RX}})^2 \cdot 2D \cdot \frac{1}{2}T_\Delta^3
\end{aligned}$$

$$\Rightarrow \sigma^2_{\Delta f_{\mathrm{err}}} = \frac{1}{(2\pi T_\Delta T_I P_{z_{RX}})^2}\left[A - 2B + C\right] = \frac{2D}{(2\pi)^2}\cdot\frac{3T_I - T_\Delta}{3T_I^2} \tag{225}$$

Since $T_I = T_a - T_\Delta$ (Equation (217)),

$$\sigma^2_{\Delta f_{\mathrm{err}}} = \frac{2D}{(2\pi)^2}\cdot\frac{3T_a - 4T_\Delta}{3(T_a - T_\Delta)^2} \tag{226}$$

B Mathematical models for radio channels

This chapter deals with the mathematical representation of static and time variant channel realizations and and also recalls some statistical measures for channel models.

B.1 Continuous-time channel representation

Due to multipath wave propagation, the radio channel behaves like a linear time-invariant (LTI) system or *filter* with transfer function $h(\tau)$ in static situations, or linear time-variant system with transfer function $h(t,\tau)$ in dynamic situations ([Joh01]). This transfer function is seen at the receiver side.

The channel is time-variant when either the transmitter, the receiver or other objects, which appear as reflectors or obstacles, are moving. Depending on the application scenario and frame size, a static channel model might be regarded as sufficient in some cases. This is valid for the frame size of the transmitted packets short enough to assume that the channel impulse response is nearly constant for the whole frame. In other cases, where this assumption does not hold, the system must be simulated with a time-variant model.

For the time-variant impulse response, [Joh01] assumes that the received signal is a pure linear combination of delayed versions of the input signal. The restriction made here is that the delayed versions of the transmitted signal are not subject to any transformation. Hence, the generic time-variant channel response can be written as a weighted sum of Dirac pulses. Each signal component is described with it's time varying delay $\tau_d(t)$ and time-varying amplitude $\alpha_d(t)$

$$h(t,\tau) = \sum_{d=1}^{N_d} \alpha_d(t)\delta(\tau - \tau_d(t)) \tag{227}$$

In (227), depending on the definition, $h(t,\tau)$ could be regarded as the transfer function arising from the gain pattern of the transmitter antenna, multipath propagation in the medium and gain pattern of the receiver antenna. The antenna gain (pattern) $G(\phi,\varphi)$, a function of the azimuth angle ϕ and elevation angle φ, is defined as the power gain in the direction (ϕ,φ) with respect to an isotropic antenna which would radiate with equal power in all directions. Due to the reciprocal nature of antennas, the same pattern applies for reception. Since each wave propagation path is associated with two angles of departure at the transmitter side and two angles of arrival at the receiver side, each path is weighted with the TX/RX antenna gains for these angles. An extended formulation of (227) distinguishes between *path loss* expressed in the amplitudes $\beta_d(t)$ and the antenna gains.

$$\begin{aligned} \alpha_d(t) &= \beta_d(t) \cdot \sqrt{G_{tx}(\phi_{tx,d}(t), \varphi_{tx,d}(t)) \cdot G_{rx}(\phi_{rx,d}(t), \varphi_{rx,d}(t))} \qquad (228) \\ \Rightarrow h(t,\tau) &= \sum_{d=1}^{N_d} \beta_d(t) \cdot \sqrt{G_{tx}(\phi_{tx,d}(t), \varphi_{tx,d}(t)) \cdot G_{rx}(\phi_{rx,d}(t), \varphi_{rx,d}(t))} \cdot \delta(\tau - \tau_d(t)) \end{aligned}$$

If the channel model applies to ideal omnidirectional antennas and includes path angle information, the model is in principle decoupled from the choice of antenna and has therefore a wider range of application. Equation (228) can then be used to obtain channel responses for any choice of specific antenna patterns. In general, the antenna for both transmitter and receiver has strong impact on the propagation characteristics. The RF signal $y(t)$ seen at the input of the receiver low noise amplifier (LNA) is given as the convolution of the transmitted signal $x(t)$ with the channel response $h(t,\tau)$. For the moment, noise contributions have been excluded.

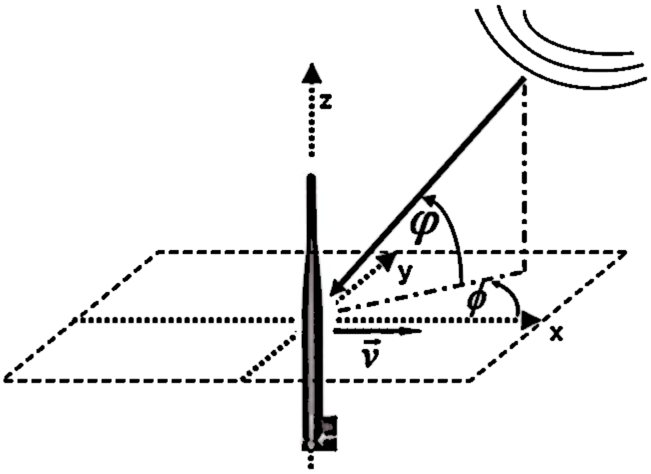


Figure 89: Definition of angles of arrival (AoA) for some wave contribution

$$y(t) = x(t) \otimes h(t,\tau) = \int_{\tau=0}^{\infty} h(t,\tau)x(t-\tau)d\tau = \sum_{d=1}^{N_d} \alpha_d(t)x(\tau - \tau_d(t)) \tag{229}$$

The transmitted and received signal in complex envelope notation read

$$x(t) = Re\{\tilde{x}(t)e^{j2\pi f_c t}\}, \quad y(t) = Re\{\tilde{y}(t)e^{j2\pi f_c t}\} \tag{230}$$

It is easy to show that for the delay-path channel (229), the complex envelope of the output signal is given by ([Joh01])

$$\tilde{y}(t) = \sum_{d=1}^{N_d} \alpha_d(t)e^{-j2\pi f_c \tau_d(t)} \cdot \tilde{x}(t - \tau_d(t)) \equiv \tilde{x}(t) \otimes \tilde{h}(t,\tau) \tag{231}$$

so that the equivalent baseband channel impulse response reads

$$\tilde{h}(t,\tau) = \sum_{d=1}^{N_d} \alpha_d(t)e^{-j2\pi f_c \tau_d(t)}\delta(\tau - \tau_d(t)) \tag{232}$$

In principle, the complex envelope of the output signal is composed in a similar way as the signal at the RF in (229), except that each signal contribution with delay τ_d is rotated by a (time-varying) phase $\phi_d(t) = -2\pi f_c \tau_d(t)$ in the complex domain. Note that there is a clear relation between the delay and the phase of the d-th filter tap of the channel response. This relation is sometimes dropped for some channel models.

We can make the convention that if the channel impulse response is supplied as a snapshot from some set of measured data, the snapshot shall correspond to $t = 0$.

B.2 Time-variant discrete-time channel representation

For simulation at sampling frequency $f_T = 1/T$, it is necessary to derive the effective discrete-time channel response which is seen by a system operating at f_T. Since the different path delays in (232) are arbitrary, resampling of the input waveform for arbitrary time shift is required. This can be accomplished with the aid of a *virtual* analog reconstruction (or Nyquist) filter after the transmitter PA. We assume that f_T is high enough for the Nyquist bandwidth $B/2$ (with $B = f_T$) to cover not only the radio signal but also the interference outside the occupied bandwidth, which is created by the transmitter PA. In addition, Doppler shifts are assumed small enough not to cause aliasing. If the reconstruction filter has passband up to $B/2$, it will not alter the signal. $g(\tau)$ shall denote the filter impulse response. The input waveform after PA is given by

$$\tilde{x}(t) = \sum_{k=-\infty}^{\infty} x_k \cdot g(t - kT) \tag{233}$$

Combining (231), (232) and (233) yields for the *sampled* output signal $y(nT)$

$$\begin{aligned}
\tilde{y}(nT) &= \tilde{x}(nT) \otimes \tilde{h}(nT, \tau) = \sum_{k=-\infty}^{\infty} x_k \cdot \int_{\tau=-\infty}^{\infty} g(nT - \tau - kT)\tilde{h}(nT, \tau) d\tau \qquad (234) \\
&= \sum_{k=-\infty}^{\infty} x_k \left(\sum_{d=1}^{N_d} \alpha_d(nT) \cdot e^{-j2\pi f_c \tau_d(nT)} \cdot g((n-k)T - \tau_d(nT)) \right) \\
&= \sum_{k=-\infty}^{\infty} x_k \left(\sum_{d=1}^{N_d} c_d(n) \cdot g_d(n, n-k) \right) \\
&= \sum_{k=-\infty}^{\infty} x_{n-k} \left(\sum_{d=1}^{N_d} c_d(n) \cdot g_d(n, k) \right) = x_k \otimes h_B(n, k) \qquad (235)
\end{aligned}$$

$$\begin{aligned}
c_d(n) &:= \alpha_d(nT) \cdot e^{-j2\pi f_c \tau_d(nT)}, \quad g_d(n, k) := g(kT - \tau_d(nT)) \qquad (236) \\
h_B(n, k) &:= \sum_{d=1}^{N_d} c_d(n) \cdot g_d(n, k)
\end{aligned}$$

$h_B(n, k)$ denotes the discrete-time band-limited impulse response of the channel. This impulse response is composed of N_d time-variant filter responses $g_d(n, k)$, which are used to realize the time-varying delays $\tau_d(nT)$. These filter responses are weighted with time-varying coeffcients $c_d(n)$. The coefficients are responsible for fast-fading and Doppler-spread.

B.3 Static discrete-time channel model

If the channel is static, then $\alpha_d(nT) \equiv \alpha_d, \tau_d(nT) = \tau_d, c_d(n) \equiv c_d, g_d(n, k) \equiv g_d(k)$,
and Equation (236) becomes

$$h_B(k) = \sum_{d=1}^{N_d} c_d \cdot g_d(k) = \sum_{d=1}^{N_d} \alpha_d \cdot e^{-j2\pi f_c \tau_d} g(kT - \tau_d) \tag{237}$$

B.4 Simplified time-variant discrete-time channel representation

For the static case, a simple discrete-time convolution must be performed. On the other hand, the dynamic case theoretically requires that the impulse response is recalculated for every output sample, a computationally expensive task. One way to reduce the computational burden is to perform a new calculation of the channel response $h_B(n,k)$ only for every L-th sample with index $n = q{\cdot}L$ and use linear interpolation to calculate the channel responses for the intermediate samples $n = q \cdot L + 1 ... q \cdot (L+1) - 1$ between two adjacent nodes $q \cdot L$ and $q \cdot (L+1)$. L must be chosen small enough to meet a required precision.

In case of a highspeed 60 GHz radio link, a further simplification seems possible for frame-based media access. It is assumed that the lowest data rate R_{min} does not fall below 100 MBit/s. As will be shown in later chapters, the maximum frame length T_{frame} can be limited to $T_{\text{frame,max}}$ = 1 millisecond without significant overhead loss and still delivers 100 kBit per frame at the lowest data rate. We may assume that the receiver is moving and the transmitter is fixed. A maximum speed of v_{max} = 5 meter/s in office situations will unlikely be exceeded. Hence, the receiver will make a step of no more than $\Delta s_{\text{max}} = v_{\text{max}} \cdot T_{\text{frame,max}}$ = 5 millimeters ahead, which is equal to one wavelength at 60 GHz.

If the receiver is at least 30 cm away from the transmitter and any reflecting surfaces, the power level of a direct line-of sight component changes only by 0.15 dB while moving in the direction of the transmitter, and the arriving angle of a ray coming from a reflector changes by no more than about one degree while moving perpendicular to this ray. In result, the path amplitudes, which depend on the angles of arrival and path lengths, will stay nearly constant during the frame.

Let's now assume that the arriving wave consists only of a single LOS component between transmitter and receiver. A step Δs in the direction of the LOS path, away from the transmitter, causes a delay change of $\Delta\tau = \Delta s/c$, c denoting the speed of light. According to (232), the delay coefficient $\Delta\tau$ leads to a phase shift of $\Delta\phi_1 = -2\pi\Delta\tau f_c$ caused by the phasor component. In addition, the signal is also retarded by $\Delta\tau$. In an OFDM system, data symbols are mapped in frequency domain on subcarriers (see 4.1). Applying an FFT in the receiver on a retarded OFDM symbol causes a phase shift of $\Delta\phi_2 = -2\pi f_{\text{max}}\Delta\tau$ for the subcarrier at the outmost frequency f_{max}. Hence, for a signal bandwidth B_{sig} of approximately $2 \cdot f_{\text{max}}$, the first phase rotation is higher by a factor of $\kappa = (\Delta\phi_1/\Delta\phi_2) = (f_c/f_{\text{max}})$. For a bandwidth of 2 GHz and a carrier frequency of 60 GHz, $\kappa = 60$, and a move of one wavelength causes a phase shift of only $360/\kappa = 6$ degree due to the retardation. Therefore, the phasor terms $e^{-j2\pi f_c \tau_d(t)}$ in (232) produce by far the largest changes of the impulse response over time. This is also true for the general case with multipath reception. Although in some rare cases the phase shifts caused by retardation may have a significant effect on the simulation result, it seems that the retardations can be neglected without falsifying the average result for a large set of channels. By neglecting the small changes of amplitudes and retardations, Equation (236) becomes

$$\begin{aligned} h_B(n,k) &= \sum_{d=1}^{N_d} \alpha_d(nT) \cdot e^{-j2\pi f_c \tau_d(nT)} \cdot g(kT - \tau_d(nT)) \\ &\approx \sum_{d=1}^{N_d} \alpha_d(0) \cdot e^{-j2\pi f_c \tau_d(nT)} \cdot g(kT - \tau_d(0)) \end{aligned} \tag{238}$$

For the simplified model, the sampled pulse waveforms $g(...)$ need to be calculated only once for the frame.

Now let's assume that the receiver is moving with constant speed v in horizontal direction in an otherwise static environment. We define that the direction is given by angle ϕ_v with respect to the positive x-axis, whereas the azimuth angle of each arriving wavefront shall be related to the negative x-axis, see Figure 89 (in the illustration, receiver direction is chosen to $\phi_v = 0$). $\tau_d(nT)$ for the d-th path contribution can be obtained from the receiver speed v and the AoA parameters (azimuth and elevation angle).

$$\begin{aligned} \phi_{rx,d}(t) &\approx \phi_{rx,d}(t=0) \equiv \phi_{rx,d} \\ \varphi_{rx,d}(t) &\approx \varphi_{rx,d}(t=0) \equiv \varphi_{rx,d} \end{aligned} \tag{239}$$

$$\tau_d(nT) = \tau_d(0) - v/c \cdot \cos(\phi_{rx,d} - \phi_v) \cos(\varphi_{rx,d}) \cdot nT \tag{240}$$

With the angular velocities

$$\Omega_d := 2\pi \cdot (f_c/f_T) \cdot (v/c) \cdot \cos(\phi_{rx,d} - \phi_v) \cos(\varphi_{rx,d}) \tag{241}$$

the time-varying channel response is finally written as

$$h_B(n,k) = \sum_{d=1}^{N_d} c_d(0) \cdot e^{j\Omega_d n} \cdot g(kT - \tau_d(0)) \tag{242}$$

B.5 Characterization of radio channels

There exist some useful measures for the description of radio channels. For some given channel response described via Equation (227), the *mean excess delay* τ_{mean} and the *rms delay spread* τ_{rms} are defined as ([Hao02])

$$\tau_{\text{mean}} = \frac{\sum_{d=1}^{N_d} P_d \tau_d}{\sum_{d=1}^{N_d} P_d} \qquad \tau_{\text{rms}} = \sqrt{\frac{\sum_{d=1}^{N_d} P_d (\tau_d - \tau_{\text{mean}})^2}{\sum_{d=1}^{N_d} P_d}} \tag{243}$$

The power-delay profile (PDP) is obtained by drawing the received power levels as a function of the delays of the associated multipath components. The mean excess delay describes where the "center of gravity" is located within the PDP of the channel. The rms delay spread as a power-weighted variance is a measure for the effective spread of the received power within the profile. The lower the rms delay spread measures, the more the multi-path contributions are concentrated around the mean excess delay. The average rms delay spread is given to

$$\tau_{\mathrm{rms,avg}} = \sqrt{E\{[\tau_{\mathrm{rms}}]^2\}} \tag{244}$$

where the average is taken over a whole set of impulse responses, and can be used to characterize a certain type of environment.

The rms delay spread can also be calculated in the discrete-time domain after sampling. The receiver sees a combined impulse response consisting of the transmitter filter, the channel and the receiver filter, $h_{tot}(\tau) = g_{\mathrm{tx}}(\tau) \otimes h(\tau) \otimes g_{\mathrm{rx}}(\tau)$. The impulse response is effectively sampled at $t_n = nT + \tau_0$ to produce $h_{\mathrm{tot}}(n) \equiv h_{\mathrm{tot}}(\tau = nT + \tau_0)$. The rms delay spread is calculated as

$$\Delta n_{\mathrm{mean}} = \frac{\sum_{n=-\infty}^{\infty} n \cdot |h_{tot}(n)|^2}{\sum_{n=-\infty}^{\infty} |h_{tot}(n)|^2} \quad \tau_{\mathrm{rms}} = T \cdot \sqrt{\frac{\sum_{n=-\infty}^{\infty} |h_{tot}(n)|^2 (n - \Delta n_{\mathrm{mean}})^2}{\sum_{n=-\infty}^{\infty} |h_{tot}(n)|^2}} \tag{245}$$

Filter responses in the transmitter and receiver add to the overall rms delay spread. The need to be taken into account when designing an OFDM system.

Another useful measure of a channel model is the *coherence bandwidth*. For its definition we need to introduce the time-variant channel frequency response[1]. It is defined as the Fourier transform of the time-variant channel response $h(t,\tau)$, performed in the delay variable τ.

$$H(t,f) = \int_{\tau=-\infty}^{+\infty} h(t,\tau) \exp(-j2\pi f\tau) d\tau \tag{246}$$

If the process is *wide-sense stationary*, the autocorrelation functions in time and frequency depend only on the time difference Δt

$$\begin{aligned} R_{hh}(\Delta t, \tau_1, \tau_2) &= E\{\overline{h(t,\tau_1)} h(t+\Delta t, \tau_2)\} \\ R_{HH}(\Delta t, f_1, f_2) &= E\{\overline{H(t,f_1)} H(t+\Delta t, f_2)\} \end{aligned} \tag{247}$$

For *uncorrelated scattering* channels, for which $R_{hh}(\Delta t, \tau_1, \tau_2) = 0$ for $\tau_1 \neq \tau_2$, it can be shown that the frequency autocorrelation function only depends on the frequency difference ([Phi63]),

$$R_{HH}(\Delta t, f_1, f_2) \equiv R_{HH}(\Delta t, f_2 - f_1) = R_{HH}(\Delta t, \Delta f) \tag{248}$$

For uncorrelated scattering, the coherence bandwidth B_α can be defined for $\Delta t = 0$ as the frequency difference, for which any two frequencies have some normalized correlation coefficient of α, typically $\alpha = 0.5$.

$$\frac{|R_{HH}(\Delta t = 0, |\Delta f| = B_\alpha)|}{|R_{HH}(\Delta t = 0, \Delta f = 0)|} = \alpha \tag{249}$$

$|R_{HH}|$ is assumed to be monotonically decreasing with $|\Delta f|$. The coherence bandwidth is a measure for frequency diversity. Frequencies with separations much greater than the coherence bandwidth are statistically independent.

---

[1] It can be shown that the usual time-frequency relationship for LTI-systems still holds ([Phi63]). For input $x(t)$ with Fourier spectrum $X(f)$ and output $y(t)$ with spectrum $Y(f)$, we have $y(t) = x(t) \otimes h(t,\tau) \Rightarrow Y(f) = X(f)H(t,f)$.

The just described class of wide-sense stationary uncorrelated scattering (WSSUS) channel models is widely used in practice ([J.D00]). For this class, the *average power delay profile* is given by

$$\mathrm{PDP_h}(\tau) = R_{hh}(\Delta t = 0, \tau_1 = \tau, \tau_2 = \tau) \equiv R_{hh}(\Delta t = 0, \tau) \tag{250}$$

It gives the average power contribution depending on the delay variable. The average PDP may consist of a continuum of delay spread or of discrete delay components, which then appear as delta-functions in (250). For WSSUS channels, the frequency autocorrelation function $R_{HH}(\Delta t = 0, \Delta f) \equiv R_{HH}(\Delta f)$ can be obtained from the power-delay profile by simple Fourier transform ([Joh01]),

$$R_{HH}(\Delta f) = \int_{\tau=-\infty}^{\infty} \mathrm{PDP_h}(\tau) \exp(-j2\pi\Delta f \tau) d\tau \tag{251}$$

Also,

$$\tau_{\mathrm{mean,avg}} = \frac{\int_{\tau=-\infty}^{\infty} \tau \mathrm{PDP_h}(\tau) d\tau}{\int_{\tau=-\infty}^{\infty} \mathrm{PDP_h}(\tau) d\tau} \qquad \tau_{\mathrm{rms,avg}} = \sqrt{\frac{\int_{\tau=-\infty}^{\infty} (\tau - \tau_{\mathrm{mean,avg}})^2 \mathrm{PDP_h}(\tau) d\tau}{\int_{\tau=-\infty}^{\infty} \mathrm{PDP_h}(\tau) d\tau}} \tag{252}$$

Hence, average delay spread τ_{rms} and coherence bandwidth B_α can be calculated from the average PDP. As an example, we recall the HIPERLAN channel models A to E introduced by ETSI ([ETS98])[1]. If treated as wide-sense stationary, the Hiperlan models fall in the class of WSSUS channels. Each model emulates a typical scenario and is specified by a discrete-time PDP of a tapped-delay-line on a time-grid of 10 nanoseconds, a reasonable simplification, since the signal bandwidth is only 20 MHz. The tabs are presumed to be uncorrelated and are treated as products of a large number of additive complex path contributions arising from *rich scattering*, i.e. they are idealized to conform to a complex Gaussian distribution, for which the amplitude follows a Rayleigh-distribution[2].

An often applied generic WSSUS model for rich scattering uses an exponentially decaying power-delay profile with Rayleigh-distributed delay-contributions. For static channels, the advantage of this model consists in its determination with a single parameter, the decaying exponent or equivalently, the RMS delay spread.

The *time variance* of the impulse response $h(t, \tau)$ leads to doppler spread of the signal spectrum, which is characterized by a doppler spectrum. In time domain, a *coherence-time* can be defined in an analogous way as the coherence bandwidth. For the time-variance, a Jakes-model is often used, where the different arriving rays at the receiver are assumed to appear with high density and are uniformly distributed over the angle of arrival. Details can be found in [Joh01] and [Ber01].

---

[2] Model D features also a Rician component for the first tap

[1] The models cover an NLOS office environment, large open space environment, both for indoor and outdoor; one out of five models handles the LOS case, all others assume a non-line-of-sight link. The channel delay spread ranges from 50 ns for model A up to 250 ns for model E.

C Mathematical basics of orthogonal frequency division multiplexing

In this section, the basics of the OFDM transmission scheme using a cyclic prefix with windowed transmission pulses is recalled in this section. It is shown how the channel response is transformed into a simple one-tap channel with a complex gain factor. In addition, the dicrete-time model is derived from the continuous time-model.

C.1 Continuous-time signal model

The synthesis equation for the continuous-time complex envelope of the transmission signal can be written as follows:

$$s(t) = \frac{1}{\sqrt{K}} \sum_{n_s=0}^{N-1} \sum_{k \in I} S_{n_s,k} g_k(t - n_s T_p), \quad T_p = T_g + T_s \tag{253}$$

$$g_k(t) = \begin{cases} \exp(j2\pi k \Delta f t), & \forall t \in [-T_g, T_s] \\ 0, & \forall t \notin [-T_g, T_s] \end{cases} \tag{254}$$

The modulation process for N OFDM symbols using K subcarriers is made up of two steps. At first, the original data bit stream is mapped in some way on complex data symbols $S_{n_s,k}$. In the second step, the transmission signal is synthesized according to (253) where each complex symbol $S_{n_s,k}$ is used as the amplitude and phase of the pulse $g_k(t - n_s T_p)$ of subcarrier k transmitted during the n_s-th OFDM symbol. The different transmission pulses $g_k(t)$ are orthogonal for some subinterval of duration T_s and represent base functions with the following property:

$$\frac{1}{T_s} \int_{t=0}^{T_s} \overline{g_l(t)} g_k(t) dt = \delta_{lk} \tag{255}$$

For this to hold, the *subcarrier spacing* Δf is chosen equal to the inverse of the analysis interval T_s.

$$\Delta f = 1/T_s \tag{256}$$

$g_0(t)$ is a rectagular pulse of length $T_p = T_g + T_s$ with a Fourier transform and power spectrum given as

$$G_0(f) = T_p \cdot \mathrm{Sinc}(fT_p), \qquad S_{g_0} = |G_0(f)|^2 / T_p = T_p \cdot \mathrm{Sinc}^2(fT_p) \tag{257}$$

The power spectrum is concentrated at a main lobe at DC and has decaying side lobes. Note from (254) that each base function $g_k(t)$ is the basic pulse $g_0(t)$ shifted by $k\Delta f$ Hz in frequency domain, so that their main energy is located at the dedicated subcarrier frequency $k\Delta f$.

I in (253) shall be defined as the index set of K active subcarriers. Usually, subcarriers around DC are omitted to avoid flicker noise and the need for DC coupling. In addition, *guard* subcarriers at the edge of the channel spectrum are also omitted to avoid interference to and from the neighbour channel.

Although the shifted signal spectra of the subcarriers do *overlap*, in principle, the receiver is able to retrieve the symbols using their orthogonality. For the moment, we assume that the original waveform is received unchanged. The receiver can perform correlation on some OFDM m_s to obtain symbol $S_{m_s,l}$ on the l-th subcarrier. From (253) and (255) we get

$$\frac{\sqrt{K}}{T_s}\int_{t=0}^{T_s}\overline{g_l(t)}s(t+m_sT_p)dt=\frac{1}{T_s}\sum_{k\in I}S_{m_s,k}\int_{t=0}^{T_s}\overline{g_l(t)}g_k(t)dt=\sum_{k\in I}S_{m_s,k}\delta_{lk}=S_{m_s,l} \tag{258}$$

We note that OFDM belongs to the class of linear modulation. The correlation performed in the receiver is equivalent to the use of a *matched-filter*, hence the signal-to-noise ratio is maximized in the presence of white Gaussian noise.

On the other hand, not all signal energy is used to retrieve the symbols. From (253), each transmit pulse actually starts earlier than the considered analysis interval. One can think of the main rectangular pulse being cyclically extended in negative time direction by a prefix of length T_g, the *guard interval*. Intentional insertion of this prefix is done to avoid ICI. This guard interval causes a decrease of performance by a factor of

$$\chi_{cp}=T_s/(T_g+T_s)=T_s/T_p \tag{259}$$

In a static multipath propagation scenario the transmitted signal is received by many paths with different amplitudes, phases and delays. This can be modelled with a convolution of the signal with an equivalent baseband channel impulse response. Assuming N_p paths in total and additive complex white noise $\eta(t)$ of spectral density N_0, the channel response and the received signal $y(t)$ can be written as

$$h(\tau)=\sum_{p=1}^{N_p}A_p\delta(\tau-\tau_p) \tag{260}$$

$$y(t)=s(t)\otimes h(\tau)+\eta(t)=\sum_{p=1}^{N_p}A_p\cdot s(t-\tau_p)+\eta(t) \tag{261}$$

where phase and amplitude are represented by complex factors A_p. We may assume that all delays τ_p are within a time window of $[0,T_g]$, i.e. the channel response is shorter than the guard time. Then we can make use of the following property for $0\le\Delta t\le T_g$:

$$\frac{1}{T_s}\int_{t=0}^{T_s}\overline{g_l(t)}g_k(t-\Delta t)dt=\exp[-j2\pi(k\Delta f)\Delta t]\delta_{lk} \tag{262}$$

Equation (262) is valid since the pulses are cyclically extended by the prefix. Hence this prefix preserves the orthogonality also for time-shifted pulses up to a maximum delay of T_g. If the receiver performs the same correlation as in (258) on some OFDM m_s to obtain the symbol on the l-th subcarrier, the outcome is given by

$$\frac{\sqrt{K}}{T_s}\int_{t=0}^{T_s}\overline{g_l(t)}y(t+m_sT_p)dt=\sum_{k\in I}S_{m_s,k}\sum_{p=1}^{N_p}\frac{A_p}{T_s}\int_{t=0}^{T_s}\overline{g_l(t)}g_k(t-\tau_p)dt+\eta_{m_s,l} \tag{263}$$

$$=\sum_{k\in I}S_{m_s,k}\sum_{p=1}^{N_p}A_p\exp[-j2\pi(k\Delta f)\tau_p]\delta_{lk}+\eta_{m_s,l}=S_{l,k}\sum_{p=1}^{N_p}A_p\exp[-j2\pi(l\Delta f)\tau_p]+\eta_{m_s,l}$$

$$=S_{m_s,l}H(l\Delta f)+\eta_{m_s,l}$$

$$\eta_{m_s,l} = \frac{1}{T_s} \int_{t=0}^{T_s} \overline{g_l(t)} \eta(t + m_s T_p) dt \tag{264}$$

Therefore, the received symbol on subcarrier l is equal to the product of the transmitted symbol times $H(l\Delta f) = \sum_{p=1}^{N_p} A_p \exp[-j2\pi(l\Delta f)\tau_p]$, which is identified as the Fourier transform of the channel response $h(\tau)$ at the subcarrier frequency $l\Delta f$.

The additional noise term $\eta_{m_s,l}$ at the correlator output has zero mean and a variance of $\sigma^2 = N_0/T_s$, which can be shown by calculating the expectation of $|\eta_{m_s,l}|^2$ and using the autocorrelation function of AWGN noise, $R_{\eta\eta}(\tau) = N_0\delta(\tau)$ ([Joh01]). The average output SNR for the l-th subcarrier is equal to

$$\mathrm{SNR}_l = E\{|H_l \cdot S_{l,k}|^2\}/(N_0/T_s) = |H_l|^2 \cdot E_{s,l,tx}/N_0 = E_{s,l,rx}/N_0 \tag{265}$$

$$E_{s,tx,l} = T_s \cdot E\{|S_l, k\}|^2, \quad E_{s,rx,l} = T_s \cdot |H_l|^2 \cdot E\{|S_l, k\}|^2 \tag{266}$$

where $E_{s,tx,l}$ is the average energy of the transmitted and $E_{s,rx,l}$ of the received pulses on subcarrier l, without counting the cyclic prefix.

In principle, the channel equalization in a coherent receiver reduces to simple phase and amplitude correction by multiplication with the inverse of the channel coefficients $H_l := H(l\Delta f)$. In practice, the receiver will use estimates $\widetilde{H}_l$ of the channel coefficients. These estimates are typically obtained with the aid of a *preamble* preceding the data part of the transmitted signal frame.

If the channel delay spread only extends up to $T_g - \Delta t$, the correlation window could start somewhere in the interval $t_{start} \in [-\Delta t, 0]$ without causing ISI or ICI. The channel response $\widetilde{h}(\tau)$ seen at the receiver is simply a delayed version of $h(\tau)$,

$$\widetilde{h}(\tau) = h(\tau - t_{start}) \tag{267}$$

C.2 OFDM pulse waveform

As already mentioned, leaving out subcarriers at the edge of the signal spectrum is a way to avoid interference to adjacent frequency channels of other subscribers. Such guard subcarriers are always needed to alleviate the requirements of the reconstruction filter following the DA-converter. An additional option consists in dropping the rectangular pulse in (254), which shows strong side lobes, in favour of a pulse, where the edges at $t = -T_g$ and $t = T_s$ are smoothed and which produces a spectrum with a faster decay. A useful generalization of the transmission pulses in (254) is given as follows

$$g_k(t) = \exp(j2\pi k\Delta f t) W(t) \tag{268}$$

$W(t)$ is a real-valued window function, which defines the amplitude envelope of the pulse and is used to describe the continuous fade-in and fade-out of the pulse shape. It is usually chosen symmetric to yield equal spectral behaviour for both edges. A common choice ([IEEa], [Jia08]) is the raised-cosine-pulse. We follow the notation in [IEEa]:

$$W(t) = \begin{cases} \sin^2(\frac{\pi}{2}\frac{t+T_g+T_{tr}/2}{T_{tr}}), & -T_g - T_{tr}/2 \leq t < -T_g + T_{tr}/2 \\ 1, & -T_g + T_{tr}/2 \leq t < T_s - T_{tr}/2 \\ \sin^2(\frac{\pi}{2}(1 - \frac{t-T_s+T_{tr}/2}{T_{tr}})), & T_s - T_{tr}/2 \leq t < T_s + T_{tr}/2 \\ 0, & else \end{cases} \tag{269}$$

This window function depends on the transition time T_{tr}, which can be chosen as a free parameter. The useful part in (269) of the pulse is the area where $W(t) = 1$. For this part, orthogonality is maintained according to Equation (263) in case of multipath reception. Hence the *effective guard interval* is reduced by T_{tr}. The introduced transition regions at $[-T_g - T_{tr}/2, -T_g + T_{tr}/2]$ and $[T_s - T_{tr}/2, T_s + T_{tr}/2]$ are shared between overlapping adjacent OFDM symbols, so that the OFDM symbol rate is kept constant.

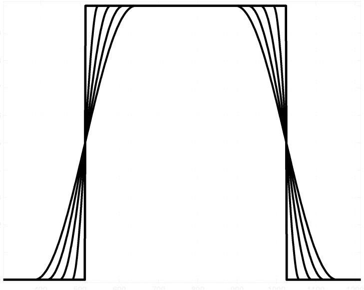


Figure 90: Raised cosine pulse for different transition durations

C.3 Discrete-time signal model

We now make the transition to a discrete-time OFDM signal model which is required both for simulation and implementation. A straight- forward approach is to sample the continuous pulse in (269) in order to attain a discrete-time representation. In a typical OFDM system, the sampling frequency is chosen as

$$f_T = 1/T = N_{FFT} \cdot \Delta f = N_{FFT}/T_s \tag{270}$$

To simplify processing, the OFDM symbol duration T_p is usually chosen as a integer multiple of the sample duration, $T_p = N_p \cdot T$. The resulting discrete-time pulses obtained from sampling (268) with the rate given in (270) read

$$g_k(nT) = \exp(j2\pi k\Delta f nT)W(nT) = \exp(j2\pi kn/N_{FFT})W(nT) \tag{271}$$

and the synthesis equation (253) becomes

$$s(nT) = \frac{1}{\sqrt{K}} \sum_{n_s=0}^{N-1} \sum_{k\in I} S_{n_s,k} \exp(j2\pi k(n - n_s N_p)/N_{FFT})W((n - n_s N_p)T) \tag{272}$$

Finally, Equation (272) can be slightly simplified by dropping the time relation T for brevity, $s(n) \equiv s(nT), W(n) \equiv W(nT)$ and also introduce additional zero symbols $S_{n_s,k} = 0$ for $k \notin I$. With the introduced zero symbols, the subcarrier index k can run over the full set of N_{FFT} subcarriers within the Nyquist bandwidth.

$$s(n) = \sum_{n_s=0}^{N-1} s_{n_s}(n) = \sum_{n_s=0}^{N-1} \frac{W(n - n_s N_p)}{\sqrt{K}} \left[\sum_{k=-N_{FFT}/2}^{N_{FFT}/2-1} S_{n_s,k} \exp(j2\pi k(n - n_s N_p)/N_{FFT}) \right] \tag{273}$$

The waveform of OFDM symbol $n_s = n_1$ with a substituted time index $n = m + n_1 N_p$ given by

$$\widetilde{s}_{n_1}(m) := s_{n_1}(m + n_1 N_p) = \frac{W(m)}{\sqrt{K}} \left[\sum_{k=-N_{FFT}/2}^{N_{FFT}/2-1} S_{n_1,k} \exp(j2\pi km/N_{FFT}) \right] \tag{274}$$

The sum within the brackets is just the offset-representation of the inverse discrete Fourier transform (IDFT) of a vector of N_{FFT} samples in frequency domain, which is transformed into a vector of N_{FFT} independent time domain samples at $m = 0...N_{FFT} - 1$.

The IDFT in the transmitter can be efficiently implemented with an inverse fast Fourier transform (FFT). Afterwards, the extended pulse waveform can be created by adding a cyclic prefix and a cyclic suffix and weighting the sample amplitudes with $W(m)$. The required number of prefix samples is given by the highest number N_{pre}, for which $W(-N_{\text{pre}}) \neq 0$. Correspondingly, the number of required suffix samples is the highest number N_{suf}, for which $W(N_{FFT} - 1 + N_{\text{suf}}) \neq 0$. As seen from (274), prefix and suffix are copies of the last N_{pre} samples and first N_{suf} samples of the IFFT block leading to cyclic extension.

Just like the transmitter is able to synthesize the main OFDM symbol block with IFFT operation, the receiver is able to perform the correlations in (258) in parallel using efficient FFT operation, i.e. using the FFT as a matched-filterbank.

The derivation (260)-(264) for the continuous-time case can also be performed in a similar way for the discrete-time case. The channel impulse response $h(\tau)$ is replaced with a discrete-time impulse response $h(n)$, and the channel transfer function (CTF) is obtained with discrete Fourier transform. Again, it can then be shown that orthogonality is preserved and that each received symbol on some subcarrier in frequency domain appears multiplied with the channel CTF at the subcarrier frequency. This relationship is valid due to the cyclic prefix, which ensures that convolution is equal to *cyclic convolution* for the samples of the analysed data block. This is the case as long as the channel is not longer than the cyclic prefix. Only cyclic convolution, like the inverse operation needed for equalization, can be handled with FFT ([Joh96]).

D Convolutional codes and decoding algorithm

D.1 Channel coding

Channel coding is an essential part of nearly every digital radio transmission system in our days. The aim of channel coding is to improve the power efficiency of the system by increasing the robustness of the transmitted data against perturbations like additive noise. The robustness is increased by adding a defined portion of redundancy to the original data packet in a controlled manner, which enables the receiver to perform error correction on the received data. The idea is well understood when taking a systematic [1] binary block code as an example, and applying hard-decision decoding in the receiver. For a block code, an arbitray large message stream is divided into blocks of k message bits (or symbols), and each block is encoded (and decoded) separately. For a binary block code, the message part consisting of k source bits is augmented by $(n-k)$ *parity bits* to produce a code word of n code bits. The parity bits are functions of the message bits. Note that only a small fraction of 2^k code words are used in a large code space of 2^n possible code words. For hard decision decoding, the code bits are first decided (mapped to zeros or ones) in the receiver and then supplied to the decoder. Depending on the channel conditions, the obtained code word may differ in some bits from the original. The decoder follows a demapping rule of assigning any possible received code word w to a message word. The optimum decoder follows the maximum *a posteriori* principle of choosing that message word m_l, which has the highest conditional probability $P(m_l|w)$ out of all messages. For the usual case where all message words have equal probability $P(m_l)$, the Bayes' theorem

$$P(m_l|w) = \frac{P(w|m_l)P(m_l)}{P(w)} \tag{275}$$

leads to the maximum-likelihood rule. According to this rule, the message word is chosen, which maximizes the likelihood $P(w|m_l)$, since $P(m_l|w) \equiv P(w|m_l) \cdot \mathrm{const.}$ Hence, the decoder assigns each message word m_l to a subset M_l of the code space, which will consist of the "valid", original error-free code word w_l and any erroneous code words, which deviate in one or a few bit positions, but are still assigned to the same message word. In this way, channel decoding can be viewed as form of *quantization* of received, erroneous code words to valid code words (and their associated message words).

In the more general case of non-systematic block codes using an alphabet of q different symbols, the encoding process defines an injective mapping from q^k message words of length k to q^k code words of length n. The decoding principle is the same.

The *Hamming distance* $d(w_1, w_2)$ is defined as the number of positions, in which two code words w_1 and w_2 differ from each other. It can be shown for the hard-decision case, that any received code word w should by corrected to a valid code word w_l, for which the Hamming distance $d(w, w_l)$ is minimized. In general, the *error detection capability* and *error correction capability* of a block code is determined *Hamming distance* properties of the code. From the classical viewpoint of code design, one tries to maximize the minimum Hamming distance $d_{\min}$, which is defined as the minimum number of symbol

---

[1] The term *systematic* specifies a code for which the code words contain the original, unaltered message ([Ber94]).

(or bit) positions, in which any two valid code words differ from each other. It is guaranteed, that a block code with minimum Hamming distance $d_{\min}$ can correct up to $t = \lfloor (d_{\min} - 1)/2) \rfloor$ bit errors.

The code rate R, $R < 1$, specifies the amount of redundancy and is defined as $R = k/n$. The code performance in terms of error correction capability can be increased by increasing the message block length k and keeping the code rate constant. In this way, the code space is enlarged and enables larger Hamming distances. The Shannon theorem states that the error probability can be made arbitrary small for large enough code words if the code rate R is below the channel capacity C ([Ber94]). The channel capacity depends on the signal-to-noise ratio.

Since the complexity of a decoder, which would just compare the received code word with any possible valid code word, rises exponentially with the code length n, only a few possible code families are used in practice. These families need to have special features in order to limit the computational complexity.

One way is to impose a high degree of *algebraic structure* on the code, which can be exploited in some way in the receiver. An example for such block codes are the powerful Reed-Solomon codes. Another promissing code family are the low-density parity check codes ([R. 62]). Low computational complexity is achieved with a sparse parity check matrix. In addition, the decoding process can be effectively performed in an iterative way.

Another big code family are the convolutional codes, which do not follow the scheme of splitting the message stream into blocks of length k. Instead, the whole message stream is encoded in one turn. As for the LDPC codes, each code bit is created from just a small set of adjacent message bits. Encoding is performed sequentially and linearly over the message stream to produce the code stream. Furthermore, the decoding process can also be done sequentially over the code stream typically applying the Viterbi algorithm ([And71]). Error correction capability is given by the *free distance* of the code.

Yet another code family are Turbo codes. Together with LDPC codes they can achieve near Shannon capacity for a moderate degree of complexity.

In general, hard decision decoding is sub-opimum in the sence that the early decision on the code bits discards information about the reliability of the received bits. A better decoding performance can achieved for convolutional, LDPC and Turbo codes when reliablity information is incorporated using *soft bits*.

The *coding gain* is defined as the save in power by applying a channel code compared to uncoded transmission and observing either an equal bit error rate (BER) or equal frame error rate (FER), for the same type of modulation. Typically, the coding gain is investigated for BPSK modulation with optimum matched filtering assuming a flat radio channel with additive white Gaussian noise as the only perturbance.

It can be shown that the bit error rate depends solely on the ratio E_b/N_0 where E_b is the energy per bit and N_0 the noise power spectral density. Note that for the same symbol rate, the code symbols need to have less power (and amplitude) in order to yield the same bit energy E_b. Although the signal-to-noise ratio is lowered for coded modulation and will produce more decision errors in case of hard-decision schemes, the error correction capability of a well-chosen channel code is able to overcompensate this effect.

D.2 Convolutional codes

A convolutional code belongs to the major class of linear codes and is created with a shift register as shown in Figure 91. We may assume that for every cycle one source bit is shifted in the register and N code bits are produced[1], so that the code rate equals $R = 1/N$.

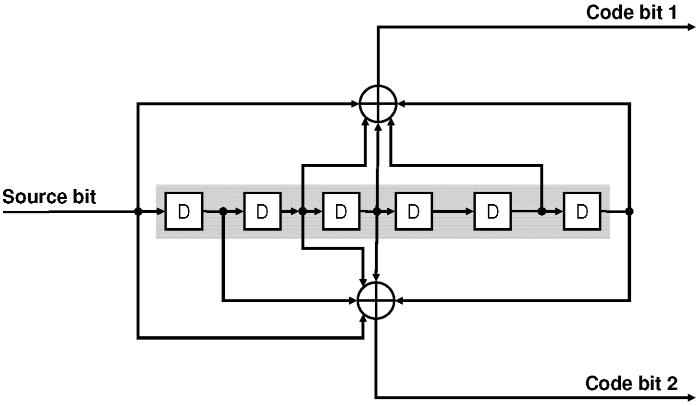


Figure 91: Encoder circuit for (133,171) convolutional code (K=7, N=2, R=1/2)

The constraint length K of the code is defined as the number of adjacent source bits involved in the computation of each code bit, which are the current input bit and the $K-1$ previous bits stored in the shift register. Therefore the length of the shift register is equal to $K-1$ leading to $M_s = 2^{K-1}$ different states. A code bit is created by summing up those bit positions, where an inteconnection exists to the dedicated output node. Modulo-2 arithmetic is used for the summation leading to xor-operation of all involded inputs. Summation is defined as $0+0=0, 0+1=1+0=1, 1+1=0$ and multiplication is defined as $0 \cdot 0 = 0, 0 \cdot 1 = 1 \cdot 0 = 0, 1 \cdot 1 = 1$. Without loss of generality, the initial state of the shift register can be fixed to be the all-zero state, as an "agreement" between encoder and decoder. This *a priori* knowledge is required for decoding.

A particular convolutional code can be described by a single binary matrix A with size $N \times K$. A 'one' in row $n \in [1, N]$ and column $k \in [1, K]$ in this matrix defines a connection from the k-th bit to the n-th output node, for the computation of the n-th code bit. The newest input bit is addressed with $k = 1$. Then the n-th output code bit $c_n(l)$ bit at information bit index l is created via convolution of the input bits $b(m)$ with the n-th row of A.

$$c_n(l) = \left[\sum_{k=1}^{K} A_{nk} \cdot b(l-(k-1)) \right]_{\mathrm{mod}2} \tag{276}$$

Although systematic codes can be constructed in this way, the best codes found are non-systematic ([Ber01]). Convolutional codes have been especially attractive since the invention of the Viterbi-decoding algorithm ([And71]), which constitutes an efficient way to perform *maximum-likelihood sequence decoding*. The condition for its proper application is that the channel is *memoryless* so that bit errors appear independently from each other.

---

[1] This is not a general restriction ([And71]), and in theory, code design for $S > 1$ source bits per cycle is an option.

We may assume that the transmitted full data packet consists of N_s message bits. This leads to 2^{N_s} different message and code bit sequences (= code words) respectively. With respect to the shift register, each message C_i also represents a unique state transition combination or "path" $(\Theta_0^{(C_i)}, \Theta_1^{(C_i)}, ..., \Theta_{N_s}^{(C_i)})$, where the states $\Theta_l^{(C_i)}$ are numbered from 0 to $M_s - 1$.

The initial state is assumed to be $\Theta_0 = 0$, as mentioned before. In addition, the final state can also be forced to be zero. This is achieved with appending of at least K *zero tail bits* to the message bits in order to clear the register. The advantage of a fixed final state will shortly become clear. Both restrictions reduce the possible number of sequences. We assume that a *branch metric* $\delta(s_1, s_2, l)$ for the possible transition from state $\Theta_{l-1} = s_1$ to state $\Theta_l = s_2$ can be defined to satisfy two conditions:

- $\delta(s_1, s_2, l)$ only depends on the N received code bits $c_n(l)$
- The sum of the branch metrics shall constitute a proper sequence metric with the property that if the probability of message C_i for the given received code word is lower than the probability of C_j, then
$$\sum_{l=1}^{N_s} \delta(\Theta_{l-1}^{(C_i)}, \Theta_l^{(C_i)}, l) < \sum_{l=1}^{N_s} \delta(\Theta_{l-1}^{(C_j)}, \Theta_l^{(C_j)}, l) \tag{277}$$

We define the accumulated branch metric $\Delta^{(C_i)}(l)$ for sequence C_i up to the l-th branch as

$$\Delta^{(C_i)}(l) = \sum_{\nu=1}^{l} \delta(\Theta_{\nu-1}^{(C_i)}, \Theta_\nu^{(C_i)}, \nu) \tag{278}$$

The Viterbi-decoding algorithm is based on the following observation: Consider the subset of sequences

$$\widetilde{C_i} \cong (..., \Theta_l, \Theta_{l+1}, ..., \Theta_{N_s}) \tag{279}$$

with some common state combination from the l-th state until the end, i.e. all these sequences remerge in state Θ_l or earlier. Apparently, these sequences differ only in the accumulated branch metrics $\Delta^{(\widetilde{C_i})}(l)$ from each other, and the decoder can select one surviving sequence $\widetilde{C_j}$ with the highest accumulated metric

$$\Delta^{(\widetilde{C_j})}(l) = \max_i \{\Delta^{(\widetilde{C_i})}(l)\} \tag{280}$$

and discard all other sequences of this subset. Note that state Θ_l can only be reached from just two previous states s_A and s_B , depending on the bit value of the rightmost flipflop in the register. This bit content represents the oldest message bit in the register and is shifted out with the next cycle (see Figure 91). Sequences, which reach state Θ_l from s_A or s_B are labeled $\widetilde{C_i}^{(A)}$ and $\widetilde{C_i}^{(B)}$ respectively. Then the maximum metric for state Θ_l can be written as

$$\max_i \{\Delta^{(\widetilde{C_i})}(l)\} = \max([\max_i \{\Delta^{(\widetilde{C_i}^{(A)})}(l-1)\} + \delta(s_A, \Theta_l, l)], [\max_i \{\Delta^{(\widetilde{C_i}^{(B)})}(l-1)\} + \delta(s_B, \Theta_l, l)]) \tag{281}$$

using the mutual max-function $\max(x, y) = x$ for $x \geq y$ and $\max(x, y) = y$ otherwise. The maximum accumulated metric for state $\Theta_l = s$ using *all sequences*, which have state $\Theta_l = s$ in common, but are not restricted to (279), shall be denoted as

$$G(l,s) := \max_{\substack{i \\ \Theta_l = s}} \left\{ \Delta^{(C_i)}(l) \right\} \tag{282}$$

Until now we have focused on some subset according to (279), but Equation (281) does not depend on specific states Θ_{l+1}, Θ_{l+2} and further, it is valid for all sequences having $\Theta_l = s$ in common. Note also that the decoder could have performed metric maximization for states A and B in a previous iteration. Hence we can formulate a recursive equation, which applies for all hypothetical states s at all time indices $l > 1$ as follows:

$$G(l,s) = \max\left[G(l-1, s_A(s)) + \delta(s_A(s), s, l), G(l-1, s_B(s)) + \delta(s_B(s), s, l)\right] \tag{283}$$

We can make the convention that state number s shall be defined as the binary shift register content as shown in Figure 91, but read from right to left. The rightmost bit, which is the oldest message bit in the register, is then interpreted as the MSB and the newest message bit as the LSB. After l interations, the encoder state number in ordinary binary notation MSB...LSB is equal to

$$s \equiv b_{l-K+1} b_{l-K+2} ... b_l \tag{284}$$

For this numbering convention, the precedent states $s_A(s)$ and $s_B(s)$, which may lead to the odd or even state s, are identified as

$$s_A(s) = \lfloor s/2 \rfloor \qquad s_B(s) = \lfloor s/2 \rfloor + M_s/2 \tag{285}$$

The decoder decision for state s at bit index l shall be defined as

$$D(l,s) = \begin{cases} 0 & \text{for} \quad G(l-1, s_A(s)) + \delta(s_A(s), s, l) > G(l-1, s_B(s)) + \delta(s_B(s), s, l) \\ 1 & \text{else} \end{cases} \tag{286}$$

The *previous state* which had the higher metric, is then obtained as

$$s_{\text{pre}}(l,s) = \lfloor s/2 \rfloor + D(l,s) \cdot M_s/2 \tag{287}$$

Equations (283), (285), (286) and (287) are the basis for decoding operation. Decoding consists of a trace-forward and a traceback procedure. During a trace-forward step, the decoder evaluates Equation (283) for all states s of some bit index l performing add-compare-select (ACS) operations. Since there remains only one survivor path for each state s at bit index l, each state metric $G(l,s)$ is identical to the accumulated path metric of the survivor. Furthermore, there are just M_s survivors left after each iteration. The decoder remembers each path decision by storing $D(l,s)$ in memory. The old path metrics $G(l-1,s)$ are not required after the l-th trace forward step and can be overwritten by the updated path metrics $G(l,s)$.

Once the trace-forward procedure is accomplished, the decoder could in theory identify the path with the highest accumulated metric and recover all information bits associated with this path using all involved path decisions $D(l,s)$. This is done by traversing back on the survivor path in accordance with

(287) and identifying the LSB of each state s_l at bit index l as the l-th information bit, following our convention (284). A very long path memory of length N_s would be required, and the identification of the final survivor also needs high computational effort. In addition, the last bits would suffer from a higher error probability. Fortunately, it can be shown that any of the final M_s paths will merge on the traceback after approximately $\Delta_m = 5 \cdot K$ backsteps with very high probability ([Joh01]). This fact can be exploited to limit the path memory and to avoid the necessity to identify the winner path. The decoder can perform trace-forward and traceback operation in an interlaced way. Each time a traceback procedure starts from some bit index l, only those bits are retrieved, which are more than Δ_m steps back in time, since these bits face a high degree of error protection. Furthermore, the traceback can always begin from the zero state $s = 0$. For the last traceback procedure starting at index $l = N_s$, a traceback depth Δ_m cannot be maintained. Therefore it is required to append the aforementioned tail bits to terminate the stream. Due to these bits, the last state is enforced to be the all-zero state, and decoding from that state is correct per definition. The tail bits also protect the last data bits from higher error rate.

On the other hand, there exist a convenient way to enforce state s=0 to be the initial state. The path metrics merely need to be initialized in that way as to set a positive offset value $V >> 0$ for state s=0 and assign a zero metric value to all other states.

The error correction properties of the code are well reflected in the decoding algorithm. If a wrong path remerges with the correct one after just a few steps $L \geq K$, and strong channel disturbance forces this erroneous path to have a higher metric, then the decoder will erroneously choose that path even if all further metric decisions will be correct.

A thorough treatment of convolutional codes featuring all aspects including visualizations in form of the state- and trellis diagram is given in [And71] and [Ber94].

Puncturing

Puncturing means to delete a certain fraction of the generated code bits in a systematic fashion in order to obtain a higher code rate. The discarded code bits are replaced by soft bits with a neutral metric of zero in the decoder. The decoder is still able to perform its job, but error correction capability is compromised. Puncturing usually happens block-wise taking a block of N_p code bits and discarding particular bits. For 802.11a, the puncturing patterns for the r=1/2 code are defined as follows.

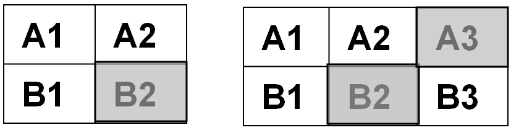


Figure 92: Standard puncturing patterns for r=1/2 convolutional code to obtain r=2/3 and r=3/4

The first row holds the code bits from the upper path in Figure 91 and the second row the code bits from the lower path. The bits at grayed-out positions are discarded. The left pattern rises the code rate to r=2/3 and the right pattern yields r=3/4.

D.3 Soft-decision bit metrics for convolutional codes

To achieve maximum coding gain, soft-decision bit metrics are required which fulfill conditions (277). We focus on OFDM transmission in frequency selective channel environments. Interleaving is required to avoid burst errors caused by channel fades if adjacent code bits would be mapped on the same subcarrier or on adjacent subcarriers. Hence each code bit $c_n(l)$ is mapped on some arbitrary subcarrier with index $I(n,l)$ exposed to a complex channel coefficient $H_{I(n,l)}$. For simplicity, the channel shall be static. The receiver may ideally have perfect channel state information, and only complex additive Gaussian noise shall disturb the symbols. We assume a constant complex noise power of $\sigma_N^2 = N_0$ before equalization. Each subcarrier symbol shall first undergo equalization so that the received constellation points match the ideal points for the corresponding modulation. The demapping from subcarrier symbols to soft-decision bit metrics is performed afterwards. In this case, the symbol associated with code bit $c_n(l)$ is affected by noise having a power of

$$\sigma_{N,n,l}^2 \equiv \sigma_{N,I(n,l)}^2 = N_0/|H_{I(n,l)}|^2 \tag{288}$$

After interleaving and mapping, the message sequence C_ν is directly associated with a complex symbol sequence

$$\vec{S^{(\nu)}} = (S_1^{(\nu)}, S_2^{(\nu)}, ..., S_{L-1}^{(\nu)}, S_L^{(\nu)}) \tag{289}$$

grouped in some order. The k-th symbol S_k shall be transmitted on subcarrier with index $J(k)$. At first, we can express the likelihood function using this symbol sequence. The received symbol sequence is denoted as

$$\vec{R^{(\nu)}} = (R_1^{(\nu)}, R_2^{(\nu)}, ..., R_{L-1}^{(\nu)}, R_L^{(\nu)}) \tag{290}$$

Then the log-likelihood function for a complex Gaussian distribution reads

$$\begin{aligned} L(\vec{R}|\vec{S}) &= \log\left[\prod_{\nu=1}^{L} P(R_\nu|S_\nu)\right] = \log\left[\prod_{\nu=1}^{L} \frac{1}{\pi\sigma_{N,J(\nu)}^2} \exp\left(-\frac{|R_\nu - S_\nu|^2}{\sigma_{N,J(\nu)}^2}\right)\right] \\ &= -\sum_{\nu=1}^{L} \frac{|R_\nu - S_\nu|^2}{\sigma_{N,J(\nu)}^2} + \text{const} = -\frac{1}{N_0}\sum_{\nu=1}^{L} |H_{J(\nu)}|^2 |R_\nu - S_\nu|^2 + \text{const} \\ &= -\frac{1}{N_0}\left[\sum_{\nu=1}^{L} |H_{J(\nu)}|^2 (\text{Re}\{R_\nu\} - \text{Re}\{S_\nu\})^2 + \sum_{\nu=1}^{L} |H_{J(\nu)}|^2 (\text{Im}\{R_\nu\} - \text{Im}\{S_\nu\})^2\right] + \text{const} \end{aligned} \tag{291}$$

From (291) it follows that if each code bit is mapped onto one dimension only, either the real or imaginary part of the symbol, then all inphase and quadrature components involved in the likelihood function are completely decoupled. Indeed, according to Figure 94, this is the case for all defined mapping rules. Hence, bit metric formulation becomes a one-dimensional problem.

Bit metrics for BPSK and QPSK for maximum likelihood decoding are easily derived. For BPSK, where only the inphase term is used, the quadrature term in (291) will give a constant noise term common to all subcarriers and can be ignored. The transmitter maps each code bit $c_n(l) \in (0;1)$ to ± 1 and assigns it to symbol $\kappa(l,n)$ after interleaving

$$S_{\kappa(l,n)} = 2 \cdot c_n(l) - 1 \tag{292}$$

Using (292), it is well known ([And71],[Fil02]) and easy to show that (291), apart from a constant term, becomes a scalar product of the BPSK transmit sequence with the received sequence. But since the channel is frequency selective, the summands in the scalar product are weighted with the power coefficients $|H_{J(\kappa(l,n))}|^2$.

$$L(\vec{R}|\vec{S}) = \frac{1}{N_0}\sum_{l=1}^{N_s}\sum_{n=1}^{N}|H_{J(\kappa(l,n))}|^2\mathrm{Re}\{R_{\kappa(l,n)}\}\cdot S_{\kappa(l,n)} + \mathrm{const} \tag{293}$$

The inphase components $R_{\kappa(l,n)}$ can be identified as the soft-decision bit metrics which have to be weighted and merely flipped in sign if the corresponding code bits are zero, $c_n(l) = 0$. We can define the weighted bit metric $m(l,n)$ for BPSK as

$$m(l,n) = |H_{J(\kappa(l,n))}|^2\cdot\mathrm{Re}\{R_{\kappa(l,n)}\} \tag{294}$$

and Equation (293) becomes

$$L(\vec{R}|\vec{S}) = \frac{1}{N_0}\sum_{l=1}^{N_s}\sum_{n=1}^{N}m(l,n)\cdot(2\cdot c_n(l)-1) + \mathrm{const} \tag{295}$$

The decoder requires branch metrics for all possible state transitions. For the standard (131,171) channel code, we have N=2 code bits per branch giving four combinations (0;0), (0;1), (1;0) and (1;1), and we can precalculate the metric combinations and store them in a temporary vector $\Psi(l,\mu)$, which is updated with new branch metrics for each iteration.

$$\begin{pmatrix}\Psi(l,1)\\ \Psi(l,2)\\ \Psi(l,3)\\ \Psi(l,4)\end{pmatrix} = \begin{pmatrix}-m(l,1)-m(l,2)\\ -m(l,1)+m(l,2)\\ +m(l,1)-m(l,2)\\ +m(l,1)+m(l,2)\end{pmatrix} \tag{296}$$

The decoder has simply to choose one of the four bit metrics $\Psi(l,\mu)$, $\mu = 1..4$ for each branch metric $\delta(s_1,s_2,l)$, depending on the values of the code bits associated with the fictive state transitions $s_1 \rightarrow s_2$.

$$\delta(s_1,s_2,l) \in \Psi(l,\mu) \tag{297}$$

This selection in (297) determined with Equation (276). As mentioned before, QPSK can be regarded as two parallel BPSK modulations performed in the inphase and quadrature-component, and the results for BPSK also apply for QPSK.

For 16-QAM and 64-QAM, the ML-function $L(\vec{R}|\vec{S})$ in (291) is still valid but of no practical use. The reason is that there is no simple way to derive bit metrics from this equation. To keep the processing chain in the same way as for BPSK and QPSK, which consists of a demapper, followed by a deinterleaver and finally the Viterbi-decoder, we can follow [Fil02] on the derivation of simplified, sub-optimum bit metrics for 16-QAM and 64-QAM modulation. As stated in [Fil02], the simplification leads to negligible performance degradation. In contrast to (291), this approach essentially relates the received symbol sequence to the code sequence by the likelihood function

$$L(\vec{R}|\vec{c}) = log(P\{\vec{R}|\vec{c}\}) = \sum_{n,l} log(P\{R_{\kappa(l,n)}|c_n(l)\}) \tag{298}$$

Either the inphase or the quadrature component of the received symbol $R_{\kappa(l,n)}$ is involved, depending on where $c_n(l)$ is mapped. In principle, the decoder performs mutual comparison of two sequences in each ACS-unit. This can be done by comparing the likelihood functions $L(\vec{R}|\vec{c}^{(i)})$ with $L(\vec{R}|\vec{c}^{(j)})$ of two message sequences C_i and C_j and selecting the one with the larger metric. On the other hand, it is of practical advantage to use different metric instead of $L(\vec{R}|\vec{c})$. We can define soft bits $m(l,n)$ as

$$m(l,n) = log(P\{R_{\kappa(l,n)}|c_n(l) = 1\}) - log(P\{R_{\kappa(l,n)}|c_n(l) = 0\}) \tag{299}$$

The soft bits can be used to form an alternative sequence metric as written below.

$$\tilde{L}(\vec{R}|\vec{c}) = \sum_{n,l} m(l,n)(2 \cdot c_n(l) - 1) \tag{300}$$

The likelihood functions of two different sequences differ only for those code bits which are different. From the basic identity $a > b \Leftrightarrow a - b > b - a$ it follows that

$$L(\vec{R}|\vec{c}^{(i)}) > L(\vec{R}|\vec{c}^{(j)}) \Leftrightarrow \tilde{L}(\vec{R}|\vec{c}^{(i)}) > \tilde{L}(\vec{R}|\vec{c}^{(j)}) \tag{301}$$

Comparing (295) and (300) reveals the advantage of this metric. The defined soft bits can be used the same way as the previously defined weighted bit metrics for BPSK and therefore, the same standard Viterbi decoder can be used for all higher modulation schemes. The conditional probabilities $R_{\kappa(l,n)}|c_n(l) = \alpha\}$, $\alpha = 0, 1$ depend on all constellation points for which $c_n(l) = \alpha$ and are in general complicated terms. [Fil02] uses a log-sum approximation to attain simplified metrics. These are shown in Figure 93 for 16-QAM and 64-QAM in the lower graphs. These graphs apply for the gray-mappings in Figure 94 in appendix E. Each bit metric is plotted against the I or Q component for the different bits, with bit 1 being the MSB. The shown values are valid for unity gain. Reliability is incorporated by weighting each metric with the corresponding power coefficient $|H_{J(\kappa(l,n))}|^2$. Note that the average reliability of the different bits is not equal, but decreases with higher bit index. This fact needs to be addressed when designing the interleaver. In general, bit metrics for M-QAM decrease in amplitude for rising modulation index M. The reason is that the constellation points must lie on a narrower grid if a constant average power is maintained for all M-QAM constellation sizes. With the given normalized soft bit metrics, it is possible to switch between different modulations within the same code stream.

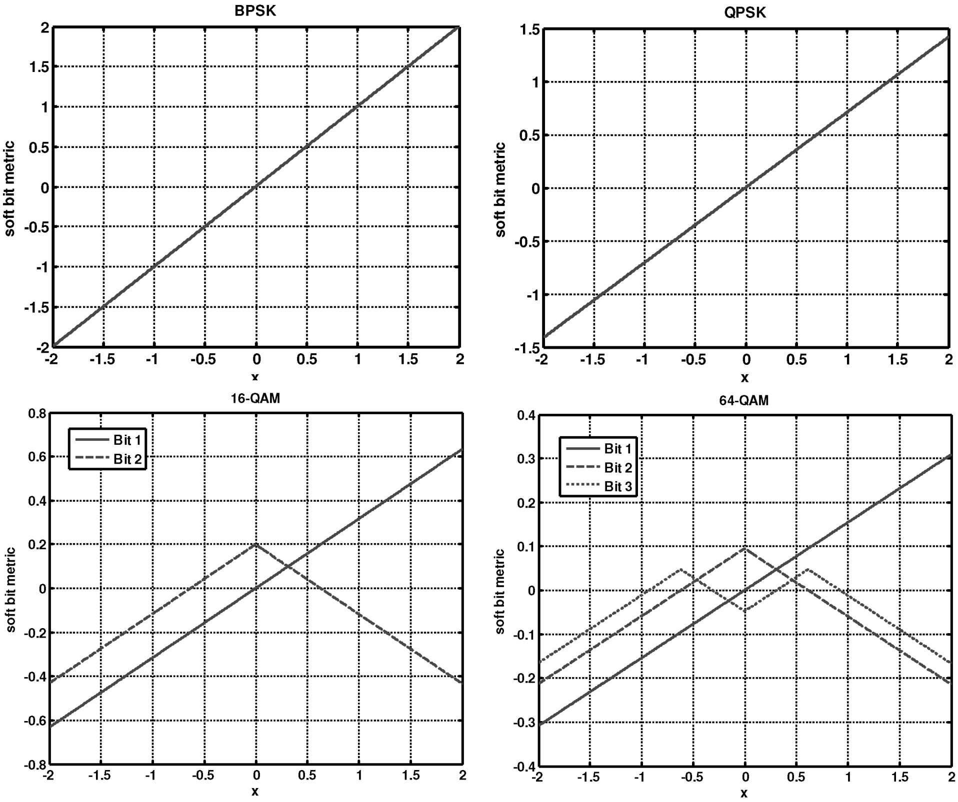


Figure 93: Bit metrics for BPSK/QPSK and simplified bit metrics for 16-QAM/64-QAM

E BPSK, QPSK and QAM-mapping with gray-encoding

Table 14: QPSK / M-QAM amplitude correction factors

BPSK	QPSK	16-QAM	64-QAM
1	$f_{QPSK} = \sqrt{1/2}$	$f_{16} = \sqrt{1/10}$	$f_{64} = \sqrt{1/42}$

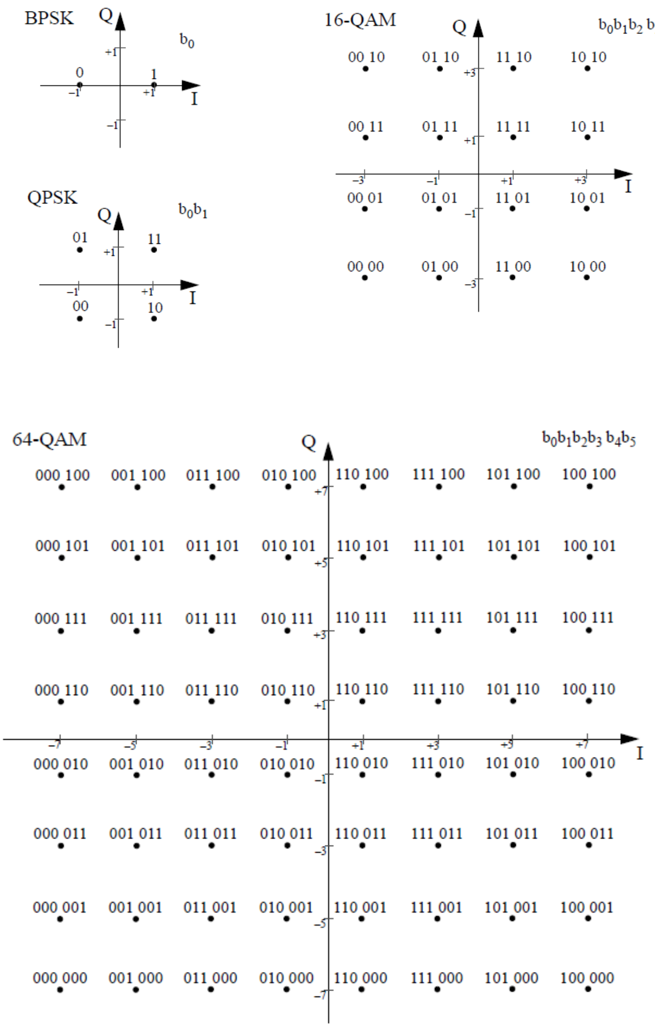


Figure 94: BPSK, QPSK 16-QAM and 64-QAM mapping using gray-encoding ([IEEa])

F Partitioning algorithm for fast Fourier transform

The FFT is the core unit in any OFDM system and contributes much to the success of the OFDM modulation scheme. The system benefits from the efficiency of the algorithm, which reduces the number of operations required to compute the discrete Fourier transform (DFT). The DFT $X(k)$ for a block of N time samples $x(n)$ is given as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \qquad 0 \leq k \leq N-1 \qquad W_N = e^{-j2\pi/N} \tag{302}$$

This can be written in matrix notation as $\vec{X} = Q_N \cdot \vec{x}$ with matrix Q_N having elements $(Q_N)_{(k+1),(n+1)} = W_N^{kn}$. This is done in order to see that direct computation would need N^2 complex multiplications.

The following short derivation of the FFT-step is based on ([Joh96]). If N can be factored as a product $N = LM$ of two integers L and M, we define $X(p,q) := X(k = Mp + q)$ for $0 \leq p \leq L-1$ and $0 \leq q \leq M-1$ and rewrite Equation (302) as

$$X(p,q) = X(k = Mp + q) = \sum_{n=0}^{N-1} x(n) W_N^{(Mp+q)n} \tag{303}$$

We can define $x(l,m) := x(n = Lm + l)$ for $0 \leq l \leq L-1$ and $0 \leq m \leq M-1$. The summation over index n can be split into two nested summations by setting $n = Lm + l$.

$$X(p,q) = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} x(Lm+l) W_N^{(Mp+q)(Lm+l)} \tag{304}$$

$$= \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} x(l,m) W_N^{MLmp} W_N^{Lmq} W_N^{Mpl} W_N^{lq}$$

$$= \sum_{l=0}^{L-1} \left\{ W_N^{lq} \left[\sum_{m=0}^{M-1} x(l,m) W_M^{mq} \right] \right\} W_L^{lp} \tag{305}$$

The last line (305) follows from the identities $W_N^{Nmp} = 1$,$W_N^{mqL} = W_{N/L}^{mq} = W_M^{mq}$ and $W_N^{Mpl} = W_{N/M}^{pl} = W_L^{pl}$. The inner summation is identified as an M-point DFT. We can label this DFT as

$$X_l(q) = \sum_{m=0}^{M-1} x(l,m) W_M^{mq} \tag{306}$$

and write (305) as

$$X(p,q) = \sum_{l=0}^{L-1} \left\{ W_N^{lq} X_l(q) \right\} W_L^{lp} \tag{307}$$

The product $W_N^{lq} X_l(q)$ has the same indices. We define

$$\hat{X}_l(q) = W_N^{lq} X_l(q) \tag{308}$$

and rewrite equation (308) as

$$X(p,q)=\sum_{l=0}^{L-1}\hat{X}_l(q)W_L^{lp} \tag{309}$$

The right side is identified as an L-point DFT. M different equations for $0\le q\le M-1$ lead to M of such DFT operations. We label these DFTs as $\widehat{X}_q(p)$ and get

$$\widehat{X}_q(p)=\sum_{l=0}^{L-1}\hat{X}_l(q)W_L^{lp} \tag{310}$$

$$X(p,q)=X(k=Mp+q)=\widehat{X}_q(p) \tag{311}$$

We can write down the elements of $x(l,m)$ in matrix form with l being the index of L rows and m the index of M columns. Since $x(l,m)=x(n=Lm+l)$, this is done column by column. Then Equation (306) suggests to perform M-point DFTs over L rows, replace the original row contents with the DFTs and use a row index q instead of m. Equation (308) suggests to multiply each element in the new matrix with W_N^{lq}. Finally, Equation (310) suggests to carry out an L-point DFT over each column q, replace the column contents with the obtained DFTs and use a column index p instead of l. The last line tells us that the coefficients must be read out row by row.

The procedure is visualized in Figure 95 for a $N=36$-point DFT. The number of multiplications is reduced from $N^2=36^2=1296$ to $2\cdot N^2+N=2\cdot 6^2+36=108$, so that the main effort is reduced by a factor of 12. But if we do not count multiplications with unity, which appear for $W_N^{kn}=1$ if $k=0$ or $n=0$, the effort for the direct computation is $36^2-36-35=1225$, and for the shortened computation we have $(36-6-5)\cdot 2+36-6-5=75$. This gives a reduction factor of 16.333.

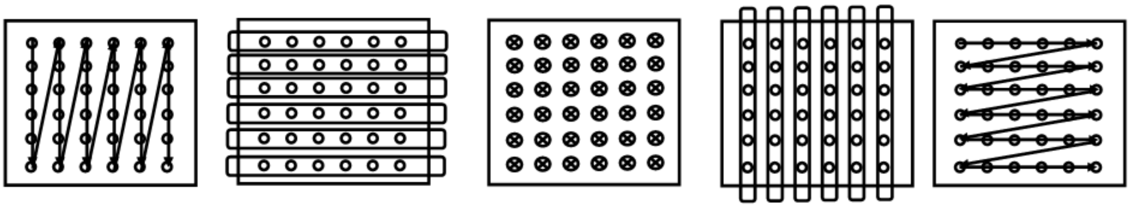


Figure 95: 36-point DFT performed with row-DFTs, factorization, and column-DFTs

G OFDM PHY parameters for narrowband and wideband mode

Table 15: PHY parameters for narrowband mode (WIGWAM)

FFT bandwidth	f_T = 400 MHz
Occupied signal bandwidth	$B_{\mathrm{sig}} \approx$ 333 MHz
FFT size	N_{FFT} = 256
Subcarrier spacing	Δ_{SC} = 1.5625 MHz
Data subcarriers	N_D = 192
Continuous Pilot subcarriers	N_P = 16
Outer guard subcarriers	N_g = 43
DC zero subcarriers	N_z = 5
Symbol duration	$T\mathrm{sym}$ = 800 ns
FFT interval	T_{FFT} = 640 ns
Cyclic prefix	T_g = 160 ns

Table 16: PHY parameters for wideband mode (EASY-A)

FFT bandwidth	f_T = 2160 MHz
Occupied signal bandwidth	$B_{\mathrm{sig}} \approx$ 1757 MHz
FFT size	N_{FFT} = 1024
Subcarrier spacing	Δ_{SC} = 2.109375 MHz
Data subcarriers	N_D = 768
Continuous Pilot subcarriers	N_P = 60
Outer guard subcarriers	N_g = 191
DC zero subcarriers	N_z = 5
Symbol duration	$T\mathrm{sym} \approx$ 592.6 ns
FFT interval	$T_{\mathrm{FFT}} \approx$ 474 ns
Cyclic prefix	$T_g \approx$ 118.5 ns

H Elementary hardware blocks

This short section presents elementary processing blocks, which found use in the FPGA-based WIGWAM demonstrator. Most blocks have been designed using fixed point arithmetic with truncation. Truncation simply consists in omitting one or a few least significant bits (LSBs) and has the advantage that no additional rounding logic is required. Fixed point numbers ν with B bits precision can be thought of as to exist in the limited interval $[-1, 1 - \mathrm{LSB}]$, $\mathrm{LSB} = 2^{-(B-1)}$.

H.1 CORDIC processors

CORDIC stands for "COordinate Rotation DIgital Computer" and refers to an number of efficient algorithms for trigonometric functions ([Jac59]). Three funtions found use in the demonstrator.

1. The calculation of the angle $\phi = \angle(z)$ of some complex number z.
2. The operation of rotating a complex number z by an angle ϕ, $\widetilde{z} = ze^{j\phi}$.
3. The operation of rotating a complex number z_2 in direction of $\phi = -\angle(z_1)$ for $|z_1| > 0$. This operation corresponds to $\widetilde{z_2} = z_2\overline{z_1}/|z_1|$.

The basic idea behind the CORDIC algorithm is to favourably make use of the property that rotation of a complex number z by a fixed angle

$$\phi_k = s_k \cdot \angle(1 + j2^{-k}), \quad s_k \in (-1, 1) \tag{312}$$

in positive or negative direction can be performed by a complex multiplication

$$\begin{aligned} \widetilde{z} &= z \cdot r_k && (313) \\ r_k &= 1 + js_k 2^{-k} && (314) \end{aligned}$$

although the length of $\widetilde{z}$ will be altered. For the particular choices r_k, this complex multiplication merely consists of two shift operations and two additions. This is an important side aspect of the idea and allows simple implementation.

$$\begin{aligned} z \cdot r_k &= (\mathrm{Re}\{z\} + j\mathrm{Im}\{z\}) \cdot (1 + js_k 2^{-k}) && (315) \\ &= (\mathrm{Re}\{z\} + j\mathrm{Im}\{z\}) - (\mathrm{Im}\{z\} - j\mathrm{Re}\{z\}) s_k 2^{-k} && (316) \end{aligned}$$

For properly selected signs s_k, the infinite sum of the angles ϕ_k will converge to any arbitrary angle ϕ in the positive half-plane,

$$\phi = \sum_{k=0}^{\infty} \phi_k \quad \phi \in [-\pi/2, \pi/2) \tag{317}$$

Hence, the rotation by an angle ϕ is achieved with

$$z_\infty = z \cdot \prod_{k=0}^{\infty} r_k \tag{318}$$

and no explicit trigonometric functions have been used. Note that

$$\lambda = \left| \prod_{k=0}^{\infty} r_k \right| \approx 1.64676 \tag{319}$$

so that z will be streched by a factor of λ, $|z_\infty| = \lambda|z|$, but this factor is constant for all selections of s_k and can be compensated. CORDIC operation can be performed in *vector mode* and *rotational mode*. We define the series of the complex numbers z_k, $k > 0$, obtained after the $(k-1)$-th step as

$$z_k = z \cdot \prod_{l=0}^{k-1} r_l \quad \Rightarrow z_k = z_{k-1} r_{k-1} \tag{320}$$

and the corresponding accumulated angle steps Φ_k, $k > 0$, as

$$\Phi_k = \Phi_0 - \sum_{l=0}^{k-1} \phi_k \qquad \Rightarrow \Phi_k = \Phi_{k-1} - \phi_k \tag{321}$$

In vector mode, we first set $z_0 = z$ and $\Phi_0 = 0$ and restrict z to lie in the positive half-plane, $\mathrm{Re}\{z\} \geq 0$. z is always rotated in direction towards the positive real axis if s_k is chosen as

$$s_k = \begin{cases} +1, & \text{for} \quad \mathrm{Im}\{z_{k-1}\} < 0 \\ -1, & \text{for} \quad \mathrm{Im}\{z_{k-1}\} \geq 0 \end{cases} \tag{322}$$

This means that z is essentially rotated by $-\angle(z)$ so that $\Phi_\infty = \angle(z)$ for $\Phi_0 = 0$ and z_∞ will align with the positive axis, $z_\infty = \mathrm{Re}\{z_\infty\} = \lambda|z|$. Therefore, apart from the elongation by factor λ, this mode performs transformation from cartesian coordinates $(\mathrm{Re}\{z\}, \mathrm{Im}\{z\})$ to polar coordinates $(z_\infty = \lambda|z|, \Phi_\infty = \angle(z))$.

To extend the range to the whole complex plane, an additional initial step is needed. A complex input number $\hat{z}$ with $\mathrm{Re}\{\hat{z}\} < 0$ must be flipped in sign before usual CORDIC computation can take place, and this is accounted for with $\Phi_0 = \pi$. Hence, the initial parameters for the extended input range are set to

$$z_0 = \begin{cases} \hat{z}, & \Phi_0 = 0 \quad \text{for} \quad \mathrm{Re}\{\hat{z}\} \geq 0 \\ -\hat{z}, & \Phi_0 = \pi, \quad \text{for} \quad \mathrm{Re}\{\hat{z}\} < 0 \end{cases} \tag{323}$$

The rotational mode can be used to rotate some vector $\hat{z}$ by a given angle of $\varphi \in [-\pi, \pi)$. In particular, if we would choose $\hat{z} = 1/\lambda$, we would obtain an output vector of $z_\infty = e^{j\varphi}$. The initial values are set to

$$z_0 = \begin{cases} \hat{z}, & \Phi_0 = \varphi \quad \text{for} \quad \varphi \in [-\pi/2, \pi/2) \\ -\hat{z}, & \Phi_0 = \varphi + \pi \quad \text{for} \quad \varphi < -\pi/2 \\ -\hat{z}, & \Phi_0 = \varphi - \pi \quad \text{for} \quad \varphi >= \pi/2 \end{cases} \tag{324}$$

This time, the signs s_k are chosen to

$$s_k = \begin{cases} +1, & \text{for} \quad \Phi_{k-1} \geq 0 \\ -1, & \text{for} \quad \Phi_{k-1} < 0 \end{cases} \tag{325}$$

According to (321) and (325), the angle Φ_k converges to $\Phi_\infty = 0$ and the output vector will indeed converge to $z_\infty = \hat{z}e^{j\varphi}$. Therefore, if we set $\hat{z} = R$ with $R \geq 0$, transformation from polar coordinates (R, φ) to cartesian coordinates $(\lambda R \cos\varphi, \lambda R \sin\varphi)$ is carried out.

For this work, a new type of CORDIC processor has been introduced. We consider the operation

$$\widetilde{z^{(2)}} = z_2 \overline{\hat{z}}/|\hat{z}| = z^{(2)} e^{-j\angle(\hat{z})} \tag{326}$$

This could be done with two consecutive CORDICs, the first one operating in cartesian, and the second one in rotational mode. A better solution is to *combine* cartesian and rotational mode into one algorithm. We consider the cartesian algorithm again. Since $\hat{z}$ is rotated by $-\angle(\hat{z})$, the same rotation can be simultaniously achieved for another complex number $z^{(2)}$, if the same shift and add operations in (320) are also applied on $z^{(2)}$, using the same decisions s_k.

$$z_k^{(2)} = z_{k-1}^{(2)} r_{k-1} \tag{327}$$

$$z_0^{(2)} = \begin{cases} z^{(2)} & \text{for} \quad \mathrm{Re}\{\hat{z}\} \geq 0 \\ -z^{(2)} & \text{for} \quad \mathrm{Re}\{\hat{z}\} < 0 \end{cases} \tag{328}$$

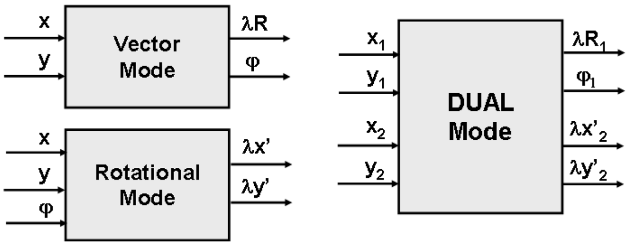


Figure 96: CORDIC processors in Vector-, rotational- and dual-mode

The three different functional blocks are illustrated in Figure 96. In a real implementation, the number of steps is restricted to some value K. In addition, the internal bitwidth B_ϕ for the angles ϕ_k and the bitwidth B_{xy} for cartesian coordinates $\mathrm{Re}\{z_k\}$ and $j\mathrm{Im}\{z_k\}$ are limited. The CORDICs have been implemented as configurable, pipelined blocks with adjustable bitwidth parameters B_{xy}, B_ϕ, step parameter K and "computational depth" N_c. The last parameter determines the number of cordic steps between two pipeline registers. For lower clock speed or bit resolution, more consecutive iterations N_c can be performed from one register stage to the next within one clock cycle.

For an input $z = 1 + j$, the absolute value $R = \lambda\sqrt{2} = \sqrt{2} \cdot 1.64676$ is larger than 2, i.e. larger than the fixed point range. Therefore, to avoid clipping, a cartesian input must be scaled down by two bit.

The down-shifted values in (315) are immediately truncated to B_{xy} bits before any further operation. Note that precision declines for lower cartesian input values. To achieve a wide dynamic range, a high internal resolution is required for cartesian coordinates.

H.2 Multipliers

Since the FPGA already contains synchronous multipliers, design of multiplication block was not required. For the Virtex-2, the multipliers have an input width of 18 bit and deliver 36 output bits with the next cycle. In the implementation, a higher precision was never required. *Complex multiplication* can be realized with four or three real multipliers.

$$(x_1 + jy_1) \cdot (x_2 + jy_2) = x_1x_2 - y_1y_2 + j(x_1y_2 + y_1x_2) \equiv A + B + j(C + D) \tag{329}$$

$$= x_1(x_2 - y_2) + y_2(x_1 - y_1) + j[x_2(x_1 + y_1) - x_1(x_2 - y_2)] \equiv \hat{A} + \hat{B} + j(\hat{C} - \hat{A}) \tag{330}$$

In most cases where timing is critical, the logical path where multiplication is performed, cannot contain additional logic. With this assumption, and if the complex multiplier is used as a synchronous block, pipeline delay is three cycles for the three-multiplier version and two cycles for the four-multiplier version. Numerical range is doubled for two real input numbers with equal range. Some LSBs can be immediately dropped mutiplication to alleviate timing requirements for the last adder stage. Due to the large amount of hardwired multipliers, the four multiplier version has been preferred for the baseband implementation. It omits the three input adders and has less latency. The complex multiplier block can be freely configured ni terms of bitwidths.

Now we shortly consider the MSBs. The output of the multipliers is already scaled down by one bit, since the product of the largest negative fixed point numbers $(-1) \cdot (-1) = 1$ would exceed the range. For optimum output range, the complex product is saturated. Since the multipliers scale down by one bit, we obtain for the product $(-1 - j)(-1 + j)/2 = 1$, and this is saturated to $1 - \mathrm{LSB}$. It is the only input combination, where saturation is needed.

H.3 $1/x$-Dividers

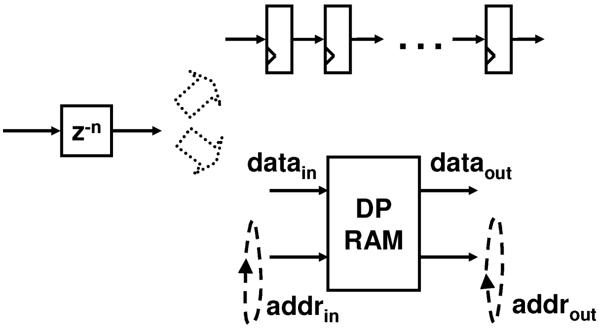
For division, the $1/x$ block for real, nonnegative numbers x has been implemented. Operation is done with the aid of a lookup-table. The $1/x$-divider is fully pipelined and has a delay of three cycles. Note that any fixed-point input number except zero is given in the form $zz...z|1|mm...$, i.e. it consists of n_z MSB zeros, $n_z \geq 0$, followed by a one and additional n_m bits, $n_m \geq 0$. At first, the divider performs fixed-point to floating point conversion by counting the MSB zeros and obtaining the exponent and mantissa. For the mantissa, the first N_m bits after the first logical one are taken. If less than N_m bits are available, the missing positions are filled with zeros. If on the other hand the exponent exceeds a maximum negative range $-E$, the exponent is replaced with $-E$ and the mantissa is set to zero. The mantissa is used as the address of the lookup-table, where values are stored for $1/x$ for $1/2 \leq x < 1$. The last stage performs floating to fixed-point conversion using a variable shifter. Division by very small numbers or zero give the same result as division by the smallest representable number, which is still correctly processed. This number is the one with the most negative exponent $-E$ and the all-zero-mantissa. The division z_0/z for complex numbers z_0, z can be realized with a dual- CORDIC, a $1/x$-divider and two real multipliers. The division is rewritten as

$$z_0/z = z_0 \cdot exp(-j\angle z)/|z| \tag{331}$$

The dual CORDIC operation performs rotation of both numbers in the direction of $\phi = -\angle z$, as required in (331). It also gives the length of $|z|$. Division by |z| is equal to multiplication of real and imaginary part of z with $1/|z|$. Since both numbers z_0 and z are stretched in length by the CORDIC operation, no length correction is needed for the ratio.

H.4 Delay elements

Delay elements z^{-N} are used to achieve a delay of a signal by N samples. They appear in signal processing blocks like moving-average operation, autocorrelation and filtering. For FPGA designs, two different delay-units are reasonable. For short delays, shift registers with n-bit data input and output can be used. For larger delays, these blocks occupy a big amount of FPGA slices. In this case, a dual-port memory block can be used. This memory block is organized like a FIFO where read and write-operations are always performed. This means that read- and write-address pointers increment each cycle and circulate non-stop through the address range. The delay is set by a positive offset between write- and read-pointer. Care must be taken for the initial situation. The memory is not cleared in case of a hardware reset or after power-on. For some blocks, the data coming out of the delay element must be ignored for some time during an initial phase after reset.



I Hardware platform

Design flow of a high-speed physical layer can be done in several ways. One way is to develop the system on an FPGA platform with the focus on final ASIC implementation. In this case, the FPGA realization is used only for functional testing and is allowed to run at a fraction of the final clock frequency.

The other approach is to realize the baseband processor with full throughput on one or several FPGAs. Since it was not planned to fabricate an ASIC, this approach has been followed. For the WIGWAM project, a commercial FPGA platform has been selected, the "FFP Basic" board from IAF company, Braunschweig ([faFGib]). A simplified block diagram is depicted in Figure 97. This board is based on two Virtex-2-pro-100 FPGA devices, which are the largest available FPGAs within the Virtex-2 family in terms of logic cells. In addition, two add-on boards are used for data conversion. Medium-size Virtex-4 SX35 FPGA devices on these boards are intented to be used to interface to the converters and for implementation of interpolation and decimation filters. A board specification summary is given in table I.

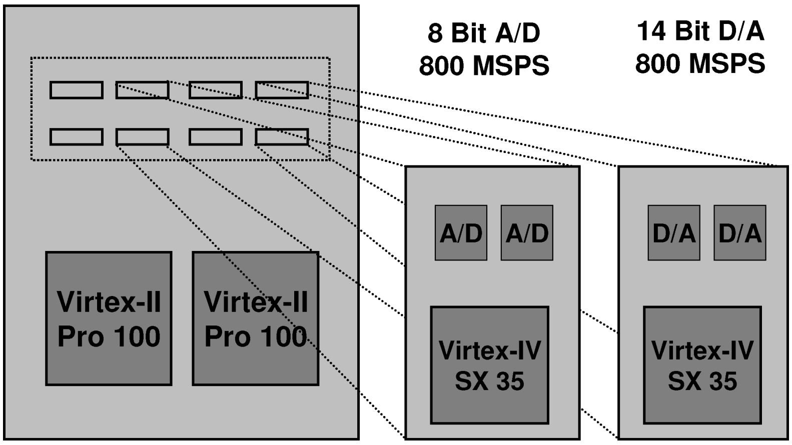


Figure 97: FPGA evaluation board with A/D and D/A converters

A logic cell in a Virtex-2-pro-100 device consists of a 4-input programmable lookup table, followed by an optional flipflop and carry logic. In addition, the FPGA features dedicated hardware multipliers and configurable memory blocks.

The Virtex-IV SX 35 device can achieve higher clock speeds, since it belongs to a newer design family and has less routing area due to smaller size. The logic cell consists of a simmilar structure. A DSP block in a Virtex-IV is made up of a 18x18 multiplier, an additional adder and accumulator [1]. In general, it appears to be a difficult task to obtain an equivalent "gate count" for the utilization of logic cells, since these cells vary in structure from one device family to the other. Details regarding FPGA field architecture can be found in [Cor04] and [Cor05].

---

[1] the often encountered multiply-and-accumulate (MAC) logic can be directly implemented this way

Table 17: Hardware platform specification

D/A-converter sampling rate	800 MSPS
D/A-converter bitwidth	14 bit
A/D-converter sampling rate	800 MHz
A/D-converter effective number of bits (ENOB)	7.2 bit
Virtex-2 pro 100 - logic cells	99216
Virtex-2 pro 100 - 18x18 multipliers	444
Virtex-2 pro 100 - 18-Kbit dual-port memory blocks	444
Virtex-4 SX35 - logic cells	34560
Virtex-4 SX35 - DSP blocks	192
Virtex-4 SX35 - 18-Kbit dual-port memory blocks	192

References

[A. 09] A. Maltsev et al. Experimental Investigations of 60 GHz WLAN Systems in Office Environment. *IEEE Journal on selected areas of communications*, 27:1488–1499, Oktober 2009.

[Ade87] Adel A. M. Saleh, Reinaldo A. Valenzuela. A Statistical Model for Indoor Multipath Propagation. *IEEE Journal of Selected Areas in Communications*, Vol. SAC-5, No. 2, Feb. 1987.

[Ahm99] Ahmad R. S. Bahai, Burton R. Saltzberg. *Mutli-Carrier Digital Communications*. Kluwer Academic, Plenum Publishers, 1999.

[Alf04] Alfonso Luís Troya Chinchilla. *Synchronization and Channel Estimation in OFDM: Algorithms for Efficient Implementation of WLAN Systems, PhD Dissertation*. 2004.

[And71] Andrew J. Viterbi. Convolutional Codes and Their Performance in Communication Systems. *IEEE Transactions on Communication Technology*, Volume 19, Issue 5, Oktober 1971.

[Beh98] Behzad Razavi. *RF Microelectronics*. Prentice Hall, 1998.

[Ber94] Bernd Friedrichs. *Kanalcodierung*. Springer, 1994.

[Ber01] Bernard Sklar, editor. *Digital Communications*. Prentice-Hall, second edition edition, 2001.

[C. 96] C. Berrou, A. Glavieux. Near Optimum Error Correcting Coding and Decoding: Turbo-Codes. *IEEE Transactions on communications*, 44:1261–1271, 1996.

[C. 09] C. Choi, M. Piz, E. Grass. Performance Evaluation of Gbps OFDM PHY Layers for 60-GHz Wireless LAN Applications. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC), Tokyo*, 2009.

[Chr91] Christoph Rapp. Effects of HPA-nonlinearity on a 4-DPSK/OFDM-signal for a digital sound broadcasting system. In *Second European Conference on Satellite Communications (ECSC-2), Liege*, Oktober 22-24, 1991.

[Cor04] XILINX Corporation. Virtex-II Pro Platform FPGAs: Introduction and Overview (DS083), March 2004.

[Cor05] Xilinx Corporation. Virtex-4 Family Overview, June 2005.

[CRY] CRYSTEK Microwave. Voltage Controlled Oscillator CVCO55CW-0400-0800. Technical report.

[D. 04] D. Costello, S. Lin. *Error Control Coding (2nd Edition).* Prentice Hall, 2004.

[Denl2] Denis Petrovic, Wolfgang Rave, Gerhard Fettweis. Performance degradation of coded-OFDM due to phase noise. Proc. of IEEE VTC, Volume 2, Issue , 22-25 April 2003 Page(s): 1168 - 1172 vol.2.

[D.G03] D.G. Brennan. Introduction to "Linear Diversity Combining Techniques" (Classic paper). *Proceedings of the IEEE,* Vol. 91, No. 2, 2003.

[DP04] G. Fettweis D. Petrovic, W. Rave. Intercarrier Interference due to Phase Noise in OFDM - Estimation and Suppression. *IEEE Vehicular Technology Conference*, 2004.

[Ena] Enablers for Ambient Services & Systems Part A - 60 GHz Broadband Links. http://www.easy-a.de.

[ETS98] ETSI EP BRAN9. 30701F: Criteria for Comparison. July 1998.

[F. 04] F. Herzel, F. Winkler, J. Borngraeber. Jitter and phase noise in oscillators and phase-locked loops. *Proceedings of the SPIE*, 5473:16–26, 2004.

[F. 05] F. Herzel, M. Piz, E. Grass. Frequency Synthesis for 60 GHz OFDM systems. *Proceedings of the 10th International OFDM Workshop, Hanburg*, 2005.

[F. 07a] F. Herzel, S. Glisic, W. Winkler. Integrated frequency synthesizer in SiGe BiCMOS technology for 60 and 20 GHz wireless applications. *Electronics Letters,* Vol. 43, No. 3, 2007.

[F. 07b] F. Yan, W.P. Zhu, O. Ahmad. Carrier Frequency Offset Estimation and I/Q Imbalance Compensation for OFDM Systems. *EURASIP Journal on Advances in Signal Processing*, Vol2007, 2007.

[faFGib] Institut für angewandte Funksystemtechnik GmbH (iaf bs.de). FFP Basic product data sheet.

[Fil02] Filippo Tosato, Paola Bisaglia. Simplified Soft-Output Demapper for Binary Interleaved COFDM with Application to HIPERLAN/2. In *IEEE International Conference on Communications (ICC), New York*, 2002.

[Fra99] Frank Herzel, Behzad Razavi. A Study of Oscillator Jitter Due to Supply and Substrate Noise. In *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol 46., No.1*, 1. January 1999.

[FUJ04] FUJITSU. MB86064 Dual 14-bit 1 GSa/s DAC, Version 1.1. Technical report, Oktober 2004.

[G. 98] G. Caire, G. Taricco, E. Biglieri. Bit-Interleaved Coded Modulation. *IEEE Transactions on Information Theory*, Vol. 44, No.3:927–946, 1998.

[Ger05] Gerhard Fettweis, D. Petrovic, M. Windisch, P. Zillmann, E. Zimmermann. Dirty RF: a new paradigm. In *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2005.

[G.F89] G.Fettweis, H. Meyr. Parallel Viterbi Algorithm Implementation: Breaking the ACS-Bottleneck. *IEEE Transactions on Communications*, Vol. 37:785–790, 1989.

[Hao02] Hao Xu, Vikas Kukshya, Theodore S. Rappaport. Spatial and Temporal Characteristics of 60-GHz Indoor Channels. *IEEE Journal of Selected Areas in Communications*, Vol. 20, No. 3, April 2002.

[Hei98] Heinrich Meyr, Marc Moeneclaey, Stefan A. Fechtel. *Digital Communication Receivers - Synchronization, Channel Estimation and Signal Processing*. Wiley Series in Telecommunications and Signal Processing, 1998.

[Her99] Hermann Rohling, Thomas May, Karsten Brüninghaus, Reiner Grünheid. Broad-Band OFDM Radio Transmission for Multimedia Applications. *Proceedings of the IEEE*, Vol. 87, No. 10, 1999.

[Hir07] Hiroshi Harada, Ryuhei Funada, Hirokazu Sawada, Shuzo Kato. CM MATLAB Release 1.1 Support Document. Technical report, IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs), Jan. 2007.

[IEEa] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz Band. *IEEE Std 802.11a-1999*.

[IEEb] IEEE 802.15 WPAN Task Group 3c (TG3c). Millimeter Wave Alternative PHY. Technical report, http://www.ieee802.org/15/pub/TG3c.html.

[IEE07] IEEE P802.15 Wireless Personal Area Networks. TG3c Selection Criteria. Technical report, IEEE 802.15.3c, January 18, 2007.

[J. 98] J. Stott. The effects of phase noise in COFDM. *EBU Technical Review*, 1998.

[J. 03a] J. Granado, A. Torralba, V. Baena-Lecuyer, J. Chavez. A new method for simultaneous fine time synchronization and frequency offset estimation in OFDM with simple hardware. *IEEE Globecom*, 2003.

[J. 03b] J. Tubbax et al. Joint Compensation of IQ imbalance and Frequency Offset in OFDM systems. *IEEE GLOBECOM*, 2003.

[J. 09] J. Luo, A. Kortke, W. Keusgen. Efficient Self-Calibration of Frequency-Dependent I/Q-Imbalance in Direct-Conversion OFDM Transceivers. *IEEE ISCAS*, 2009.

[Jac59] Jack E. Volder. The CORDIC Trigonometric Computing Technique. *IRE Transactions on Electronic Computers*, September 1959.

[J.D00] J.D.Parsons. *The Mobile Radio Propagation Channel*. 2000.

[J.H97] J.Hübner, S.Zeisberg, K.Koora, J.Borowski, A.Finger. Simple channel model for 60 GHz indoor wireless LAN design based on complex wideband measurements. *IEEE Vehicular Technology Conference*, May 1997.

[Jia08] Jian Luo, Wilhelm Keusgen, Andreas Kortke. Optimization of Time Domain Windowing and Guardband Size for Cellular OFDM Systems. *IEEE 68th Vehicular Technology Conference (VTC2008-Fall), Calgary, Canada*, Sept. 21-24 2008.

[Joh96] John G. Proakis, Dimitris G. Manolakis. *Digital Signal Processing*. Prentice Hall, third edition edition, 1996.

[Joh01] John G. Proakis, editor. *Digital Communications*. McGraw-Hill, fourth edition edition, 2001.

[K. 07] K. Maharatna, A.Troya, M.Krstic, E.Grass. On the implementation of a Low-Power IEEE 802.11a Compliant Viterbi Decoder. *19th International Conference on VLSI Design held jointly with 5th International Conference on Embedded Systems Design (VLSID 06)*, 2007.

[Luc98] Luciano Tomba. On the Effect of Wiener Phase Noise in OFDM Systems. *IEEE Transactions on Communications*, Vol 46., No 5., May 1998.

[M. 99] M. Speth, S. A. Fechtel, G. Fock, Heinrich Meyr. Optimum Receiver Design for Wireless Borad-Band Systems using OFDM - Part I. *IEEE Transactions on Communications*, Vol. 47, No. 11, November 1999.

[M. 04] M. Windisch, G. Fettweis. Standard-Independent I/Q Imbalance Compensation in OFDM Direct-Conversion Receivers. *9th International OFDM Workshop*, 2004.

[M. 07] M. Piz, E. Grass. A synchronization scheme for OFDM-based 60 GHz WPANs. *The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2007.

[M. 09] M. Peter, W. Keusgen. Analysis and Comparison of Indoor Wideband Radio Channels at 5 and 60 GHz. *3rd European Conference on Antennas and Propagation*, 2009.

[M. 10] M. Marinkovic, M.Piz, C.C. Choi, G. Panic, M. Ehrig, E. Grass. Performance Evaluation of Channel Coding for Gbps 60-GHz OFDM-based Wireless Communications. *PIMRC*, 2010.

[Mar05] Marcus Windisch, G. Fettweis. Preamble Design for an Efficient I/Q Imbalance Compensation in OFDM Direct-Conversion Receivers. *10th International OFDM Workshop, Hamburg*, 2005.

[Max06] Maxim Piz, Eckhard Grass. A simple OFDM physical layer for short-range high data rate transmission at 60 GHz. *11th International OFDM Workshop, Hamburg Harburg*, 2006.

[Max09a] Maxim Piz, E. Grass, M. Marinkovic, R. Kraemer. Next-Generation wireless OFDM system for 60-GHz short-range communication at a data rate of 2.6 GBit/s. *14th International OFDM Workshop, Hamburg*, 2009.

[Max09b] Maxim Piz, Milos Krstic, Marcus Ehrig, Eckhard Grass. An OFDM Baseband Receiver for Short-Range Communication at 60 GHz. *IEEE International Symposium on Cicuits and Systems (ISCAS)*, 2009.

[M.J07] M.J.Canet et al. Time synchronization for the IEEE 802.11a/g WLAN standard. *International Symposium on Personal, Indoor and Mobile Radio Communications*, Athens, 2007.

[M.K08] M.Krstic, M.Piz, E.Grass. 60 GHz Datapath Processor for 1 Gbit/s. *VLSI-SOC*, Oct 13-15, 2008.

[Nat05] National Semiconductor. ADC08D1000 High Performance, Low Power, Dual 8-Bit, 1 GSPS A/D Converter. Technical report, May 2005.

[Ode76] Odenwalder, J. P. *Error Control Coding Handbook*. Linkabit Corporation, San Diego, 1976.

[P.H94] P.H.Moose. A Technique for Orthogonal Frequency Division Multiplexing Frequency Offset Correction. *IEEE Transactions on Communications*, Vol. 42, No. 10:2908–2914, 1994.

[Phi63] Philip A. Bello. Characterization of Randomly Time-Variant Linear Channels. *IEEE Transactions on Communication Systems*, Vol. 11, Issue 4:360–393, Dec. 1963.

[R. 62] R. G. Gallager. Low-Density Parity-Check Codes. *IRE Transactions on Information Theory*, Volume 8, Issue 1:pages 21–28, Jan. 1962.

[R.H99] R.H. Walden. Analog-to-Digital Converter Survey and Analysis. *IEEE Journal on selected areas in communications*, 17, 1999.

[Ric00] Richard van Nee, Ramjee Prasad. *OFDM for Wireless Multimedia Communications*. Artech House Publishers, 2000.

[Shi03] Shinsuke Hara, Ramjee Prasad. *Multicarrier Techniques for 4G Mobile Communications*. Artech House, 2003.

[Ste93] Steven M. Kay. *Statistical Signal Processing*, volume Volume 1. Prentice-Hall, 1993.

[T. 94] T. Pollet, P. Spruyt, M. Moeneclaey. The BER Performance of OFDM Systems using Non-Synchronized Sampling. *IEEE Global Telecommunications Conference (GLOBECOM)*, 1994.

[T. 95] T. Pollet, M. van Bladel, M. Moeneclaey. BER Sensitivity of OFDM systems to Carrier Frequency Offset and Wiener Phase Noise. *IEEE Transactions on Communications*, Vol. 43, No. 2/3/4, Feb/March/April 1995.

[T. 97] T. M. Schmidl, Donald C. Cox. Robust Frequency and Timing Synchronization for OFDM. *IEEE Transactions on Communications*, Vol.45, No 12, Dec. 1997.

[Val01] Valkama et al. Advanced Methods for I/Q Imbalance Compensation in Communication Receivers. *IEEE Transactions on Signal Processing*, Vol. 49:2335–2344, 2001.

[W. 04] W. Winkler, J. Borngräber, B. Heinemann, F. Herzel, R.F. Scholz. High-frequency low-noise amplifiers and low-jitter oscillators in sige:c bicmos technology. *SPIE International Symposium on Fluctuations and Noise*, 2004.

[WiGa] WiGig Alliance. http://wirelessgigabitalliance.org/.

[WIGb] WIGWAM - Wireless Gigabit With Advanced Multimedia Support . http://www.wigwam-project.de/.

[WiM05] *WiMAX Standard, IEEE P802.16e*. IEEE 802.16e, Okt. 2005.

[Wir] Wireless HD. http://www.wirelesshd.org/. Technical report.

[Wir09] Wireless HD Alliance. WirelessHD Specification Overview. 2009.

List of Figures

1 Transceiver model 7
2 Different realizations for the random excess phase $\phi(t)$ according to Wiener model . . 10
3 AM/AM-conversion according to Rapp's model 17
4 IBO versus OBO for Rapp amplifier model and Rayleigh-distributed input amplitude . 19
5 Equivalent baseband link model 22
6 AM-AM and AM-PM characteristics of the utilized IHP SiGe BiCMOS power amplifier 24
7 Comparison between measurement and the Rapp model (continuous line) with p = 1.4 24
8 60 GHz Vivaldi antenna used for the demonstrator 27
9 CDF of rms delay spread for HHI impulse responses downsampled to 400 MHz bandwidth 29
10 CDF of received signal level for 400 and 800 MHz bandwidth (HHI model) 29
11 The modified Saleh-Valenzuela (SV) model applied to 60 GHz channels 30
12 CDF and delay spread distributions for a receiver omni-antenna and 2.16 GHz bandwidth 32
13 CDF and delay spread distributions for RX high-gain antenna and 2.16 GHz bandwidth 33
14 Overlapping power spectra of OFDM subcarriers 35
15 Channel dispersion inside and outside the guard interval 37
16 Sample leakage 38
17 Simulated versus predicted channel SIR for CM3.2 and 118 ns guard time 40
18 Power delay profile of a particular CM3.2 channel realization $h(m)$ 40
19 SIR prediction performance for varying frame timing Δn for channel $h(m)$ 40
20 BER versus Symbol SNR degradation for uncoded 16-QAM due to carrier frequency offset 42
21 BER versus symbol SNR degradation for uncoded 16-QAM due to Wiener phase noise 43
22 Schematic view of a integer-N charge-pump PLL ([F. 05]) 45
23 PLL output phase noise ([F. 05]) 45
24 simplified channel model for CFO estimation analysis 47
25 RMS frequency estimation error due to Wiener phase noise 50
26 Simulated vs. calculated carrier frequency estimation error in AWGN channel 51
27 CFO error for phase noise and AWGN. 52
28 Outage probability for frequency error 53
29 Interference statistics for $T_g = 118.5$ ns and $T_{FFT} = 474.1$ ns 59
30 Signal spectrum of narrowband PHY (WIGWAM) 61
31 Signal spectrum of wideband PHY (EASY-A) with and without windowing 61
32 Exemplary TDMA-based stream arrangement for 4 parallel code streams 63
33 Determination of number of streams 64
34 Determination for RS outer encoding of last block 65
35 General EASY-A frame structure including preamble and midambles 65
36 convolutional code performance with standard interleaver in omni-Vivaldi link scenario 69
37 convolutional code performance with standard interleaver in omni-omni link scenario . 69
38 Third permutation performed over regions for $n_{\text{fold}} = 3$ 70

39 Folded cycle group $n_{\mathrm{fold}} = 2$ ([Max09a]) 71
40 Interleaver comparison for CM2.3 and EASY-A PHY parameters ([Max09a]) 71
41 Transmitter signal flow for LDPC encoded blocks for 16-QAM 72
42 LDPC versus RS-convolutional code scheme for 16-QAM r=1/2 in CM2.3 ([Max09a]) 73
43 Amplitude waveform of typical PHY frame for wideband OFDM mode 76
44 Cross-correlation with short sequence A for wideband mode 76
45 Cross-correlation with long sequence B for wideband mode 76
46 Inverted structure of first preamble (EASY-A PHY) 77
47 Preamble processing overview for wideband mode 78
48 Short ACF and long ACF operating on first preamble 79
49 Short and long autocorrelation signals, anti-phase peaks 83
50 Inverted preamble structure 84
51 Timing adjustment and noise-reduction filtering 85
52 Smoothing filter response 86
53 Channel estimation SNR with and without the smoothing filter (as published in [Max09a]) 87
54 DFT window prior to timing adjustment for flat channel 90
55 Fine synchronization performance for TG3c channels without noise 91
56 Structure of inner receiver for wideband PHY 96
57 480 MBit/s data mode performance in HHI Omni-Vivaldi LOS channel 100
58 480 MBit/s data mode performance for HHI NLOS channels 101
59 Synchronization performance for CM2.2 102
60 Synchronization performance for CM3.2 103
61 Timing synchronization performance for CM3.1 103
62 Wideband performance for all modes except BPSK-1/2 104
63 FER performance 105
64 Achieved data rates 106
65 Transmitter baseband architecture 108
66 Receiver baseband processor architecture 109
67 Receiver input stage 110
68 Receiver scheduling with real timings 112
69 Synchronizer block diagram 113
70 Autocorrelator for 4 input ports and 1:4 rate reduction 115
71 Antiphase detector 116
72 Clustering Logic 117
73 Channel estimator and phase unwrapper 118
74 Redesigned scheme with smoothing filter 119
75 Mixed-mode CORDIC element 119
76 Pilot symbol processor 121
77 Data equalizer 123
78 Radix-4 FFT 124

79 Simplified schematic of de-interleaver 126
80 Preprocessing stage and one out of 64 ACS units 128
81 Regular flow chart of traceback processing units for continuous input 129
82 Viterbi decoder block diagram 130
83 Viterbi master controller 131
84 Traceback Unit 132
85 Feed-Forward and Feedback AGC topology 133
86 Generic AGC block diagram 135
87 Outage probability for 1 dB gain error for narrowband (left) and wideband mode (right)) 137
88 AGC settling for SNR = 40 dB 138
89 Definition of angles of arrival (AoA) for some wave contribution 145
90 Raised cosine pulse for different transition durations 154
91 Encoder circuit for (133,171) convolutional code (K=7, N=2, R=1/2) 158
92 Standard puncturing patterns for r=1/2 convolutional code to obtain r=2/3 and r=3/4 . . 161
93 Bit metrics for BPSK/QPSK and simplified bit metrics for 16-QAM/64-QAM 165
94 BPSK, QPSK 16-QAM and 64-QAM mapping using gray-encoding ([IEEa]) 166
95 36-point DFT performed with row-DFTs, factorization, and column-DFTs 168
96 CORDIC processors in Vector-, rotational- and dual-mode 172
97 FPGA evaluation board with A/D and D/A converters 175

List of Tables

2 Measured phase noise values of IHP oscillators 24
3 Measured channel data 28
4 RMS delay spread of HHI channel for 400 MHz bandwidth 28
5 Selected TG3c channel models 31
6 Chosen RX antenna beam widths for selected TG3c channels 33
7 PAPR of short and long preamble sequences for narrowband and wideband OFDM mode 75
8 Detection parameters 82
9 Coefficients of smoothing filter 86
10 Link conditions and baseband mode 100
11 Link conditions and baseband mode 105
12 AGC parameters 136
13 Calculated rise and fall times for 0.5 dB error 137
14 QPSK / M-QAM amplitude correction factors 166
15 PHY parameters for narrowband mode (WIGWAM) 169
16 PHY parameters for wideband mode (EASY-A) 169
17 Hardware platform specification 176